Editor's Note: This paper seeks to understand and validate connectivism theory. The paper questions connectivism's principles, compares it with other learning theories, and validates it in relationship to Artificial Intelligence and Artificial Neural Networks.

Does Artificial Neural Network Support Connectivism's Assumptions?

Alaa A. AlDahdouh
University of Minho, Portugal

Abstract

Connectivism was presented as a learning theory for the digital age and connectivists claim that recent developments in Artificial Intelligence (AI) and, more specifically, Artificial Neural Network (ANN) support their assumptions of knowledge connectivity. Yet, very little has been done to investigate this brave allegation. Does the advancement in artificial neural network studies support connectivism's assumptions? And if yes, to what extent? This paper addresses the aforementioned question by tackling the core concepts of ANN and matching them with connectivist's assumptions. The study employed the qualitative content analysis approach where the researcher started with purposely selected and relatively small content samples in connectivism and ANN literature. The results revealed that ANN partially supports connectivism's assumptions but this does not mean that other learning theories such as behaviorism and constructivism are not supported as well. The findings enlighten our understanding of connectivism and where it may be applied.

Keywords: learning theory; connectivism; constructivism; behaviorism; artificial neural network; ANN; neural network; artificial intelligence; AI; machine learning; e-learning; online learning; distance learning

Introduction

In 2005, George Siemens started his proposed learning theory, connectivism, by asserting the huge impact of technology on our learning activities. In his words, "technology has reorganized how we live, how we communicate, and how we learn" (Siemens, 2005, p. 1). It thus follows that learning theory should reflect these changes. Based on this assumption, he criticized former learning theories such as behaviorism and constructivism; and advocated new theoretical framework. The suggested framework, of course, incorporates technology in its principles. Specifically, one of connectivism's principles states that "Learning may reside in non-human appliances" (Siemens, 2005, p. 5). This principle alludes to the ability of technology to learn.

Before long, Stephen Downes embraced the theory and integrated it to the idea of connective knowledge where knowledge is distributed and it does not exist in a specific and single place (Downes, 2005, 2006). Downes (2012) concentrated on network structure of the internet and how it may empower online students to do things that were hardly ever possible before, such as distance collaboration and information searching. More recently, Downes shared a sequence of posts on artificial intelligence and neural network findings in indication of their relevance to connectivism.

Aldahdouh, Osório and Caires (2015) explained the idea of networked knowledge thoroughly and made it clearer in relation to AI. They claimed that connectivism is based on network science principles. Their step-by-step explanation of knowledge network (neural, conceptual and external) has led them to argue that connectivism "has been drawn from a long history of Artificial Intelligence findings" (p. 17).

On the other hand, these allegations have brought criticisms to connectivism. Verhagen (2006) argued that machine learning, inductive learning, and fuzzy logic software have nothing to do with human learning. "Modern cognitive tools are nothing but an extension of the toolkit" (Verhagen, 2006, p. 4). In Verhagen's perspective, artificial neural network does not differ from a pocket calculator. Moreover, Bell (2011) repeatedly mentioned that connectivism's principles lack of rigor and are, in most part, untested.

Although these criticisms were presented right after proposing connectivism, very little has been done to examine the relationship between AI findings and connectivism's principles. The question of whether connectivism was built on top of AI findings remained almost intact. In this article, ANN was selected to represent machine learning (which is a branch of AI) for many reasons. First, the core idea of ANN is inspired from human brain and, second, connectivists frequently refer to ANN as if ANN has supported their claims.

This study reviewed literature of both connectivism and ANN and tried to match their principles. The paper starts with brief description of connectivism. Then it moves to describe ANN concepts in relation to connectivism's assumptions. Subsections include artificial neuron, network architectures and learning algorithm. The paper avoids presenting complex algorithms to improve text clarity and readability. In addition, it avoids going into ANN details that will not serve the purpose of this study; the reader should be aware of the extendibility of concepts presented here.

Connectivism

The reasons that make educators keen to develop new learning theory can be summarized in three broad motives: (1) international growth in internet usage and the gap between learners' and school's activities (Bell, 2010; Brabazon, 2016); (2) half-life of knowledge becomes shorter and knowledge changes rapidly (Aldahdouh & Osório, 2016; Aldahdouh et al., 2015; Downes, 2006; Siemens, 2006); (3) and human-technology interaction where the interaction leads to changes in both sides of the equation; as technology changes, human also changes (Dirckinck-Holmfield, Jones, & Lindström, 2009; Siemens, 2005).

Connectivism started from these premises and integrated principles from different theories including chaos, network, and self-organization theories. Siemens (2005) introduced connectivism as an alternative theory which was not built on previous learning theories. In general, connectivism's principles are shifting focus from content itself to the connections of content. Maintaining existing connections, making decision of which connections to add, and creating new connections between different fields are essential part of learning according to connectivism.

The core assumption of connectivism is that knowledge has a structure; and this structure is better to be conceived as a network (Aldahdouh et al., 2015; Downes, 2005, 2006; Siemens, 2006). A network is a group of nodes linked to each other by connections. A node can be in one of three different levels: neural, conceptual, and external (Aldahdouh et al., 2015; Siemens & Tittenberger, 2009). A connection serves as a bridge that conveys information from one node to another. The more connections a network has, the more robust it will be. Without those connections, the whole network will fall apart. Thus, Siemens (2006) has concluded that "The pipe is more important than the content within the pipe" (p. 32). This makes the content less important in comparison to the connection. A more extreme view toward content sees it as something from the past; 'Content is a print concept' (Cormier, 2016). However, Aldahdouh et al. (2015) adopted a moderate view toward content in that "The information needs a connection to reach the target and the connection needs the flow of information to stay alive. Therefore, no flow of information exists without connection and no connection remains without flow of information" (p. 11).

The second core assumption of connectivism is that a single connection between two nodes does not have meaning in its own. The meaning is distributed across group of connections called pattern (Downes, 2006). The pattern "refers to a set of connections appearing together as a single whole" (Aldahdouh et al., 2015, p. 5). This pattern should be considered as the smallest unit that has meaning in its own. And hence, the network can be seen as a group of patterns that interact with each other to give the meaning of the entire network. Since connections may 'die' or 'live', the patterns and the knowledge are conceived as dynamic objects where some nodes become isolated and others become connected. The patterns change rapidly which made Aldahdouh et al. (2015) see knowledge as a "jellied creature" (p. 15).

The aforementioned assumptions can lead us directly to the definition of learning in connectivism. If knowledge has a network structure and meaning is distributed in dynamic patterns, then learning should be defined as "a continuous process of network exploration and patterns finding; it is a process of patterns' recognition" (Aldahdouh et al., 2015, p. 14).

This paper concentrates on one of connectivism’s principles which refers to the ability of technology to learn. This principle has been criticized and its meaning remains unclear for some researchers while others have questioned its validity (Bell, 2011; Kop & Hill, 2008; Verhagen, 2006). The paper does not provide comprehensive review of connectivism literature. The reader is recommended to see the work of Aldahdouh et al. (2015) for clearer explanation of the theory. Nevertheless, the paper will return to connectivism’s principles as it explains the concepts of ANN.

Artificial Neural Network

Artificial Intelligence (AI) refers to the art of creating machines that are able to think and act like humans; *or* think and act reasonably (Russell & Norvig, 2010). In order to build an agent that can think and act as so, the agent must be able to *learn* new things. To *learn* means that the agent should improve its performance on future tasks taking its past experience into account (Russell & Norvig, 2010). Making an agent able to learn is an area of study called Machine Learning (ML).

Artificial Neural Network or ANN is a software structure developed and based on concepts inspired by biological functions of brain; it aims at creating machines able to learn like human (Goodfellow, Bengio, & Courville, 2016; Nielsen, 2015; Russell & Norvig, 2010). Thus, ANN is part of ML. Interestingly, ANN has many other names in AI field including parallel distributed processing, neural computation and connectionism (Russell & Norvig, 2010). Most ANN types are *supervised learning* network. That is, both an input and the correct output should be given to a network where the network should learn a function that maps inputs to outputs. There are some types of ANN such as Deep Belief Network (DBN) which can do *unsupervised* and *semi-supervised learning* (Nielsen, 2015). However, research is still conducting on DBN to improve its performance. This article concentrates on supervised learning networks which showed a very good performance in wide variety of tasks.

Before proceeding into details, it is important to know that ANN is a vivid research area. Recent years, namely from 2011 to 2015, have witnessed a sequence of records breaking in the field of ML driven by ANN (Nielsen, 2015). Even more, Goodfellow et al. (2016) indicated that ANN evolves rapidly so that new best architecture "is announced every few weeks to months" (p. 331). This makes writing this article a very challenging task.

Artificial Neuron

Since a structure of ANN has been inspired by biological brain, ANN should consist of a collection of neurons. AI researchers designed artificial neurons called perceptron and sigmoid which are believed to have similar function to a biological neuron (Goodfellow et al., 2016;

Nielsen, 2015). Artificial neuron is hereafter referred to as *neuron* for short. A neuron is a node that receives input from preceding neurons and makes a decision to 'fire' to the next neurons. To make that decision, it should first evaluate each input according to its own perspective and then sum all inputs up to get a single and holistic view. Finally, a neuron presents the holistic view to its internal judgment system to make a decision to fire or not.

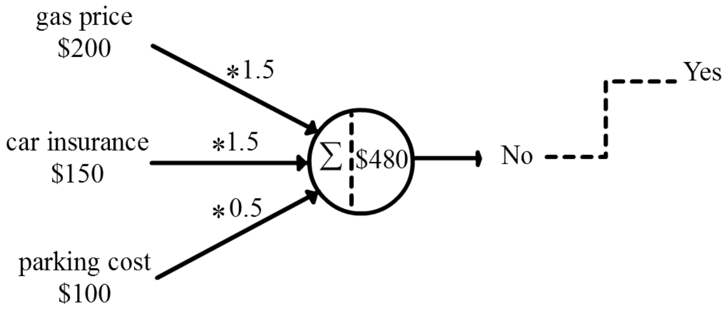

Fig. 1 Perceptron neuron

This system seems trivial but it turns out to be a complicated decision-making model. For example, suppose that you are a neuron and you want to make a decision to buy car. You probably make that decision based on many variables which may include gas price ($200), car insurance ($150), and parking cost ($100). In your perspective, car insurance and gas price are more important and more likely to increase in near future than parking cost. In this case, you weigh up car insurance (1.5) and gas price (1.5) while downplay parking cost (0.5). Then you sum that up to get the holistic perspective (100*0.5 + 1.5*150 + 1.5*200). Therefore, according to your own perspective, a car would cost you $575 per month. Then you present this holistic perspective to your internal judgment system which may have been previously set on a specific threshold ($480). Therefore, you make a decision not to buy a car because it exceeds the threshold ($575 > $480). Your own perspectives of inputs, the internal judgment system, and the threshold are called *weights*, *activation function* and *bias* respectively. By changing weights and bias you reach a completely different decision. For example, set gas weight to 1 instead of 1.5 and notice the difference. Searching for weights and bias that generate the desired output is the job of learning algorithm.

In the previous example, we imagined a neuron as a person. This sounds familiar in connectivism literature. Connectivists often argue that networks exist everywhere and these networks are similar in some way or another (Downes, 2016). Interestingly, researchers in ANN share the same assumption with connectivists (Nielsen, 2015). They sometimes conceive a single neuron as a person and in some other times conceive the whole network as a single human brain. Zooming in and out help them understand and modify both the neuron and the network in very similar way. We will see in next how this assumption works well in both levels. Another thing that matches connectivism well is the bias of a neuron. Each neuron has its own bias and, therefore, ANN contains a variety of neurons in which each neuron works in completely different way. In connectivism, Downes (2010) identifies four principles for applying democracy in education: autonomy, diversity, openness and interactivity. Siemens (2005, 2006) states that knowledge and learning rests in diversity of options and Aldahdouh et al. (2015) argues that educational systems should foster the learners' diversity, not their similarity.

Previously given example seemed as though a neuron works well in simulating human decision-making system. However, with little thinking, one can figure out that it does not. Suppose for instance that you kept all variable values as they were in the previous example except your perspective of (weight of) gas price. If you set the weight to be (1), then your holistic perspective becomes ($475). This is below your bias value ($480), thus you decide to buy a car. Now, try to set your perspective of gas to be (1.05). Your holistic perspective becomes ($485). That is greater than your bias value ($480), thus you decide not to buy a car. This is really a naive system. Our internal judgment systems do not do that. In real world, 5 dollars below or above a predefined threshold may not make that difference. This is called perceptron neuron which has a hard activation function (Russell & Norvig, 2010). It would be better if a neuron has a soft activation function. Soft so that it goes *gradually* from (Yes, I will absolutely buy a car) to (No, I will absolutely not buy a car). Note that we have just concentrated on the meaning of the output itself but it is even not logical that a very *tiny shift* in a single weight (from 1 to 1.05) makes that big difference in the output. If ANN is going to learn the right weights and biases, a small change in one weight or bias should produce small change in network output (Nielsen, 2015). In order to be able to learn, ANN should move slowly and smoothly from one decision to another. For these reasons, ANN researchers have examined many alternative soft activation functions such as *sigmoid*, *tanh* and *rectified linear* neuron.

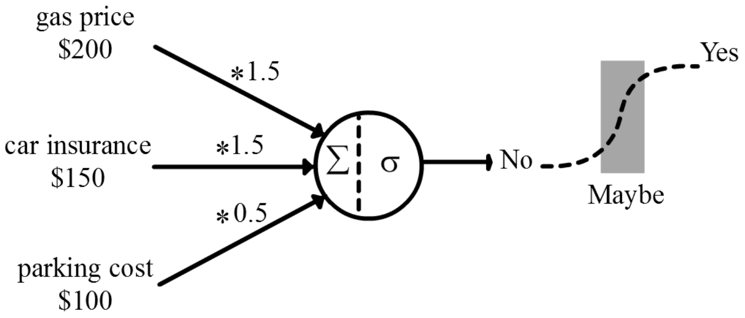

Fig. 2 Sigmoid neuron

In comparison, connectivism has been criticized for its oversimplification of interaction between nodes as the connection can be either active or inactive (Clarà & Barberà, 2014). However, connectivism proponents (Aldahdouh et al., 2015) have shown that a connection is graded and not necessarily sharp. The graded view of a connection is congruent with sigmoid neuron function. Other issue presented here is the speed of learning. Connectivism puts the bulk of its attention on the rapid change of knowledge but it does not describe how exactly a learning process is in this dynamic environment. However, there are signs in connectivism literature that a learning process should cope with this rapid change. For example, one of the connectivism's principles states that:

> Decision-making is itself a learning process. Choosing what to learn and the meaning of incoming information is seen through the lens of a shifting reality. While there is a right answer now, it may be wrong tomorrow due to alterations in the information climate affecting the decision (Siemens, 2005, p. 5).

Aldahdouh et al. (2015) emphasized the same concept and criticized the education system as they said, "Sciences are developing very rapidly and the (reluctant) derivers' decisions are coming too late" (p. 13).

Thus, the output of soft neuron goes smoothly from 0 to 1. The output is now a real number. Even though this soft neuron seems to work better than hard neuron, it still has its own problems. Take for example a sigmoid neuron.

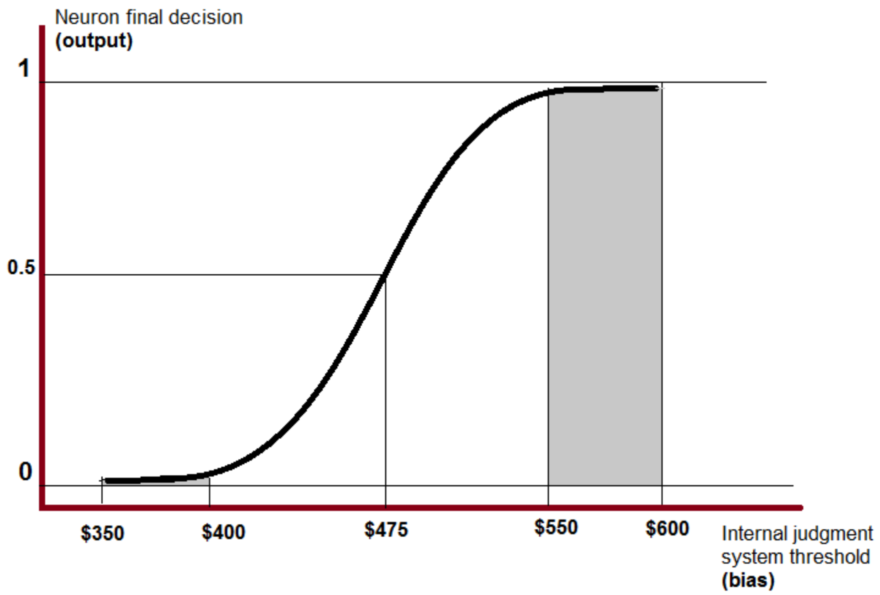

Fig. 3 Sigmoid activation function

Suppose that your initial thought was to buy a car only if it costs you less than $600 per month. That is, your bias was set to $600. Suppose further that you were wrong and you should learn that $475 is the right threshold using a sigmoid function as shown in Fig. 3. Since you learn according to a sigmoid function, you should go slowly from $600 down a curve to $475. Note that as your bias changes from $600 to $550, your decision remains nearly constant; it is almost 1. The same has to be said if you should go from $350 to $400. In these periods, the output of the neuron is saturated (Nielsen, 2015). It seems as though a neuron does not learn anything; bias changes but that does not produce a change in a decision. This is one of the problems of sigmoid neuron. To solve this issue, researchers make learning speed independent on activation function (Nielsen, 2015). Sigmoid neuron is still one of the most used neuron types in ANN. Other common soft neuron types show similar and, in some cases, better performance than sigmoid (Goodfellow et al., 2016; Nielsen, 2015). Different soft neuron types and how they reflect to educational context deserve a separate work.

An obvious and legitimate question to ask is whether these functions reflect human internal judgment system. For example, suppose a learner in an online learning setting has internal judgment system like a sigmoid. In this case, when a student decides to share and comment all the time or decides to be silent all the time, could that be a sign that he is at saturation level? Could that be a sign that he is not learning new things? Wang, Chen and Anderson (2014) indicated that "the interaction of connectivist learning is divided into four levels: operation interaction, wayfinding interaction, sensemaking interaction, and innovation interaction" (p. 121). These levels grade from concrete to abstract and from surface to deep cognitive engagement. Pattern recognition, decision making, aggregation and sharing appear on sensemaking level. This means that sharing activity resides in the same level as learning process in connectivism. Downes (2009) asserted the same concept and suggested four steps for network creation: aggregation, remix, repurpose and feed forward. Thus, educating students to share information frequently is one of the connectivism aims and it is part of student's learning process. This is not congruent with at least a flat part of sigmoid function.

Nielsen (2015) has shown that a neuron can also implement any logic gate (e.g. NOT, AND, OR and NAND). This means a single neuron has some sort of logic in its own and a network of neurons is actually a network of logic gates. For example, if you want to design a network to compute a sum of two bits, you may need a network of (n) neurons. And if you want to multiply two bits, you may need a network of (m) neurons, and so on.

From this, we have a hunch that meaning exists in pattern of connections which is one of the main connectivism's assumptions. But for now, a thing we are sure about is that one node in a network has trivial meaning in comparison to the meaning of a group of nodes. Actually, we are usually not interested in a single node meaning (AND, OR, NAND); we are interested in the meaning of the group of nodes as a whole (summation, multiplication). In connectivism, this matches the concept of emergent property (Downes, 2016) where "a compounded node is larger than the sum of its inner nodes" (Aldahdouh et al., 2015, p. 12).

Artificial Neural Network Architectures

The previous section describes a single neuron function. It is time now to see how researchers arrange group of neurons to form a learnable network. In this article, the way of arranging neurons in certain order is called *network architecture*. Recall that ANN may refer to two levels of abstraction: (1) ANN as a person's brain and (2) ANN as a group of learners. Thus, network architecture refers first to a learner's inner abilities and mental capacities and; second, refers to a way in which designers of learning-environment arrange a network of learners.

It is worth noting that ANN is a universal modeling system. Universality means that ANN can learn any given function no matter what neuron type is used. It has been proved that with few neurons and by changing biases and weights only, ANN can compute any zigzag-shaped function (Nielsen, 2015). The question now is how we arrange neurons in ANN to make it easier for a learning algorithm to find those biases and weights. For clarity and simplicity, the paper divides the most common ANN architectures based on three criteria: (1) number of layers, (2) flow of information and (3) neuron connectivity.

Number of layers:

By looking on how many layers a network has, ANN can be divided into (1) *shallow* and (2) *deep networks*.

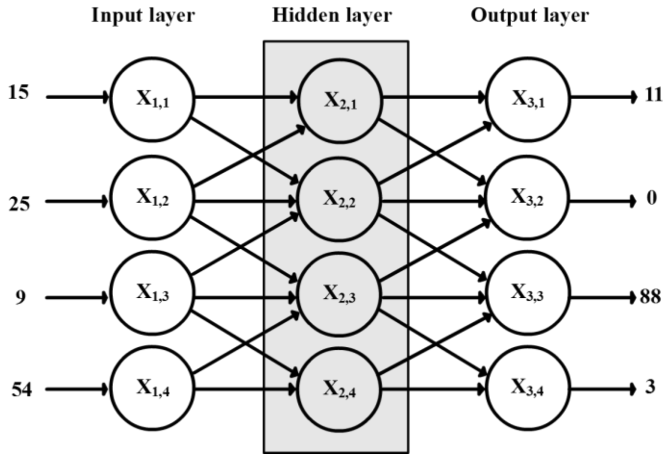

Fig. 4 Shallow neural network

A *shallow neural network* consists of three layers ordered from left to right: (1) input, (2) hidden and (3) output layer. The input layer does not really consist of neurons. Actually, it carries the input values to the network. For example, a value that passes from $X_{1,2}$ to next neurons is 25, which is the input value. The second layer is named 'hidden' because it resides in the middle and does not appear in either the input or the output of the network. Other than that, it is a normal neural layer which contains normal neurons (Nielsen, 2015). The output layer also contains normal neurons and its output represents the output of the network.

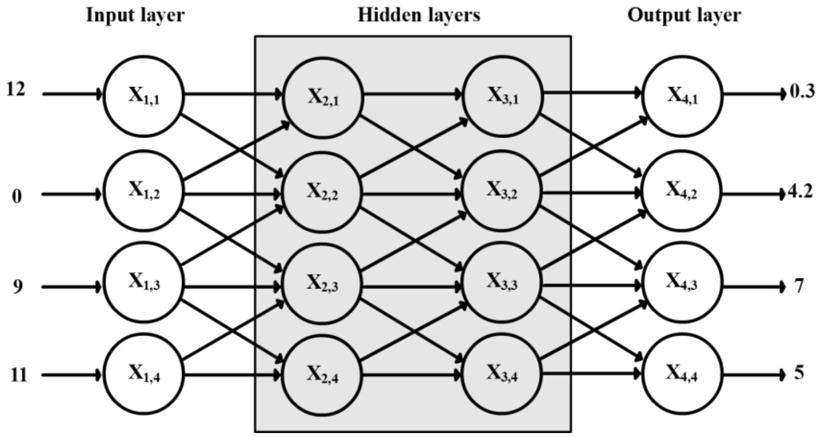

Fig. 5 Deep neural network

A *deep neural network* is the same as shallow neural network but it has two or more hidden layers. This architecture is also called Multilayer Perceptron (MLP). The original thought of presenting deep network stems from the idea of complex problem defragmentation. ANN researchers first noted that people are usually splitting the problem into sub-problems, solving each sub-problem alone and then reconstructing them to solve the entire problem (Nielsen, 2015). They inferred that if the first hidden layer is going to handle the first level of the problem, then there should be second, third and more hidden layers to handle next levels of the problem. The initial steps of training deep network were frustrating because the network took long time to train and didn't show a big difference from shallow network results.

In general, the terms *shallow* and *deep* are somehow misleading because they are not in line with educational terminology of *surface* and *deep learning* (Vermunt & Vermetten, 2004). Actually, there is nothing special in deep neural network except it gives more accurate results, if it was trained well. Moreover, the concept of *layers* is completely incompatible with connectivism's assumptions. The idea of that a network consists of a sequence of layers contradicts with chaos theory which is one of the underpinning theories of connectivism. One can argue that organizing neurons in layers is a matter of rearranging neurons positions spatially and this does not impose any constraint on neurons connectivity. This is not true, even though, because by arranging neurons in layers a neuron output is not allowed to connect to neurons in any layer other than the next layer. It should be understandable, however, that ANN researchers thought to arrange ANN in layers to facilitate the computational model of a network where each layer is represented by two mathematical *vectors*, one for biases and another for weights.

Flow of information:

By looking on how information flows through a network, ANN can be divided into (1) *feedforward* and (2) *recurrent networks*.

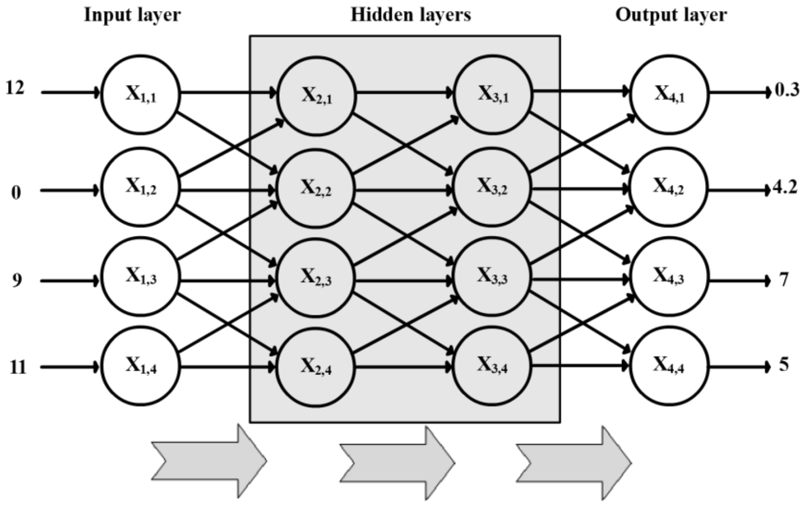

Fig. 6 Feedforward neural network

In *feedforward networks*, the output of a layer is used as an input for the next layer. There are no loops in feedforward networks; information flows in one direction where the output of a neuron can never return to its input. Feedforward network is one of the most used network structures. The value of this structure is self-explanatory since it significantly reduces the network complexity.

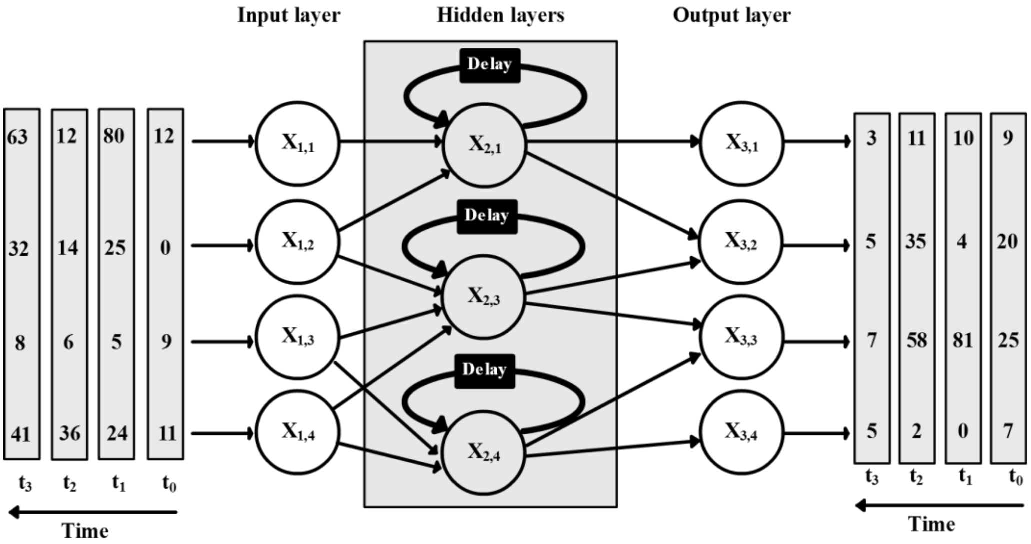

Fig. 7 Recurrent neural network

Recurrent network is a family of neural networks that processes the input sequentially and allows feedback connections (Goodfellow et al., 2016). Feedforward network structure assumes that all inputs are independent of each other (Britz, 2015). It assumes that inputs order has no meaning. This, however, turns out to be false assumption for some tasks. For example, in natural language processing, the order of words makes a significant difference in meaning. Recurrent network tries to recover this issue by allowing feedback in a network. The feedback is allowed but with a delay constraint. That is, if the *inputs* are a sequence of A, B and C; then the output of *hidden layer* in step A can only be passed to the input of the hidden layer in step B, not the hidden layer in step A itself. To make a network simple, ANN researchers usually unfold the loop to see what it looks like on each step of the inputs. In Fig. 8, one can see that a loop allows information to flow from one step to another, and, therefore, acts as a memory (Britz, 2015; Goodfellow et al., 2016; Olah, 2015).

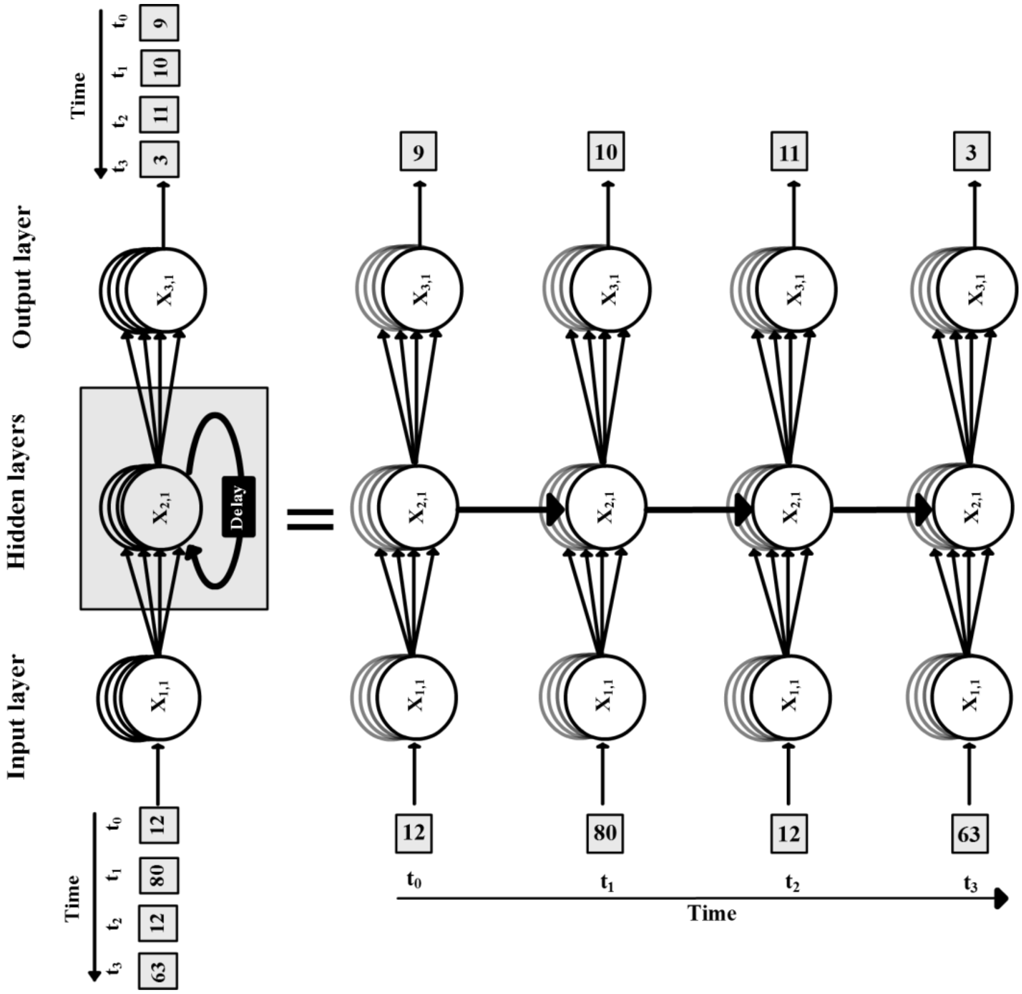

Fig. 8 Unfolded recurrent neural network

Note that at t_0, the input of neuron ($X_{1,1}$) is 12 and the output of neuron ($X_{3,1}$) is 9. At t_2, the *same neuron* ($X_{1,1}$) receives the same input (12) but the neuron ($X_{3,1}$) shows 11 as an output. This is mainly because the hidden neuron ($X_{2,1}$) receives feedback from its previous state at t_1.

Flow of information and connection directionality are some of subjects discussed in connectivism literature. Aldahdouh et al. (2015) showed that some connections in knowledge network are bidirectional while others unidirectional. They also showed that "The node can connect to itself" (p. 5). A latter concept is congruent with recurrent but not with feedforward network. However, caution should be taken when comparing ANN architecture with connectivism. Researchers restrict the flow of information in feedforward network and delay the feedback in recurrent network because it is the only way they can control and compute the output; not because they believe it is the right way of controlling the flow of information. With that said, one can consider ANN structure as a special case of connectivism network structure. The second and very important point to make here is the inclusion of time in network design. The time makes a significant difference in network output and that is one of the common points with connectivism. Connectivism's principle of shifting reality matches the example given above. "While there is a right answer now, it may be wrong tomorrow due to alterations in the information climate affecting the decision" (Siemens, 2005, p. 5). Moreover, Aldahdouh et al. (2015) clearly called for considering the time as one of knowledge dimensions.

Neuron connectivity:

By looking on how each neuron connects to other neurons, ANN can be divided into (1) *fully connected* and (2) *convolutional networks*.

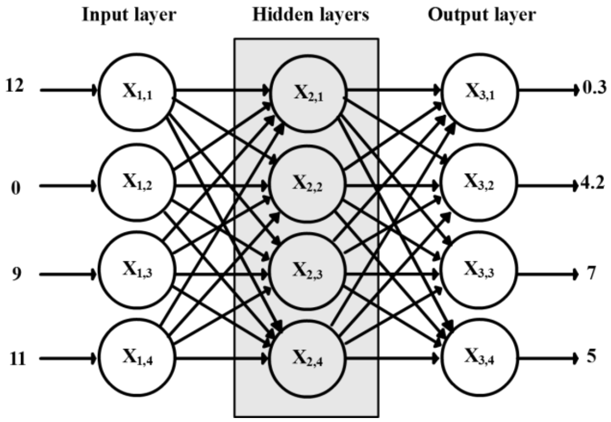

Fig. 9 Fully connected neural network

In *fully connected network,* each neuron in a specific layer is connected to all neurons in the next layer. The idea of this connectivity is to allow maximum interactions between neurons. It is also logical to think of fully connected network since we don't know in advance which connections should be removed and which ones should be remained. It is the job of learning algorithm to detect those connections. For example, if the connection between $X_{1,2}$ and $X_{2,1}$ should be removed in order to generate the desired output, the learning algorithm should figure out that the weight of this connection is 0. In other words, the learning algorithm should kill this link. One may wonder why would killing a specific connection generate a desired output? Recall the car example and how you downplayed the weight of *parking cost*. In some cases, you may even need to ignore the input at all; for example, you may need to ignore *traffic fine* as a monthly cost of a car. Full connectivity may add potentiality to the network but it adds severe difficulty on learning algorithm as well. Adding tens of neurons to fully connected network increases the number of weights and biases to be learned dramatically. Try to add two neurons to the hidden layer in Fig. 9 and notice how many new weights are added.

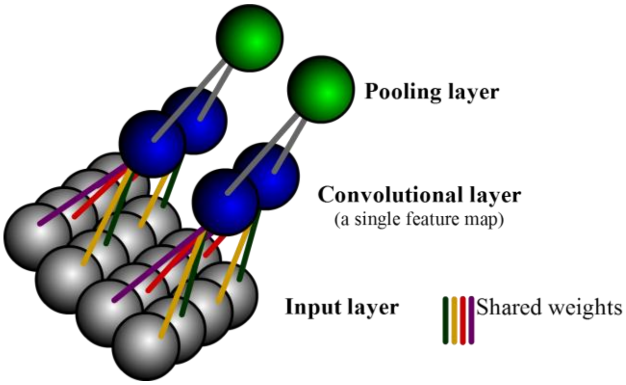

Fig. 10 Convolutional neural network

A convolutional network limits the connectivity between neurons so that a neuron in specific layer is connected only to a set of spatially adjacent neurons in the previous layer (Goodfellow et al., 2016; Olah, 2014). Moreover, neurons in the convolutional layer should weigh up the corresponding input neurons with the same values. In Fig. 10, the same color connections between the input and convolutional layer should have the same value. Those connections are called shared weights. The output of convolutional layer is often called a feature map. It is called so because when you arrange a layer as described, the output would be detecting a single feature in the input (see Goodfellow et al., 2016 for details). For example, if the input layer represents an image, a convolutional layer may detect a vertical line in that image. A convolutional layer is usually followed by a pooling layer. A pooling layer takes a feature map and tries to summarize it. For example, if a feature map detected a vertical line in a tiny spot of the image, the pooling layer would summarize that in a larger region and says: there is a vertical line in this region. The assumptions of convolutional network sound weird and complicated. From where did those assumptions come? Actually, "Convolutional networks are perhaps the greatest success story of biologically inspired artificial intelligence" (Goodfellow et al., 2016, p. 364). A convolutional network was designed to capture the same functionality of the primary visual cortex in the brain.

Connectivism appreciates network connectivity and seeks to increase it as much as possible. Three out of eight connectivism’s principles refer directly to the value of the connection (Siemens, 2005). Actually, connectivism defines learning as the process of connecting nodes in a network (Aldahdouh et al., 2015; Siemens, 2005). This may indicate that connectivism aims to make a learner as a node in the fully connected network. However, it has been proved that increasing connectivity adds complexity to ANN. This complexity makes learning harder and slower. A convolutional network, on the other hand, decreases the connectivity and achieves better results. Connectivists should pay attention to this because it disagrees with their main network designs (see Downes, 2010a work). In short, one can argue that connectivism agrees with fully connected network but disagrees with convolutional network.

It is important to note that the classification shown above is superficial and AI researchers are used to mixing network architectures together. For example, a network could be deep fully-connected feedforward network or deep convolutional network. Sometimes, a network architecture and its opposite can be mixed together. For example, a deep network may consist of two convolutional layers followed by one fully-connected layer. In general, mixing different network architectures shows better result and accuracy. In connectivism context, this may indicate that mixing connectivist's network structure (fully connected network) with other limited and loosely-connected structures would give us better educational results.

Learning Algorithm

Designing network architectures is a difficult task but training and teaching these networks are surely more difficult. To understand how ANN has been trained, it is better to start with a very simple one neuron example (Goodfellow et al., 2016; Nielsen, 2015). The principles which are used to teach a single neuron are also used to teach a whole network. However, a network level adds extra complexity which requires an additional step.

Suppose you have a very simple neuron with one input and one output. You want to teach this neuron to do a certain task (for example to memorize a multiplication table for number 5). To teach this neuron, ANN researchers usually give it a so-called *training set*. A training set contains a number of different input values (1, 2, 3, 4, 5, 6 ...) paired with the correct output (5, 10, 15, 20, 25, 30 ...).

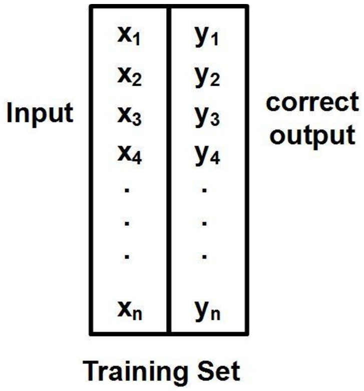

Fig. 11 Labeled training data

In the beginning, the neuron receives input and generates output according to its own weight and bias which were *randomly* selected. This means, the output of the neuron (*a*s) would most probably differ from the correct output (*y*s).

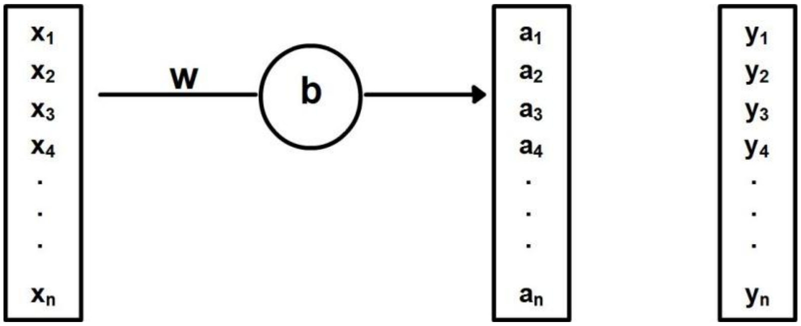

Fig. 12 Single neuron training

The difference between the neuron output and the desired output presents something useful. The function which measures the difference is often called a *cost* or *loss* function. There are many ways to calculate the cost function. One of the simplest cost functions is the Mean Squared Error (MSE):

$$C(w,b)=\frac{1}{2n}\sum_{x}\| y_x - a_x \|^2 \tag{1}$$

MSE is the average of square of differences between the correct output (y_x) and the output of the neuron (a_x) for each given input x in the training set. One may simply understand this function as a way to measure the difference between all *a*s and corresponding *y*s. It is also important to note that a cost function is written as a function of weight (*w*) and bias (*b*). That is to say, this is a cost of setting the weight and bias of the neuron in specific values. If we change the values of *w* and *b*, then we should re-calculate the cost again. If the new cost was lower, this means that the difference between the desired output (*y*s) and the neuron output (*a*s) became smaller. That is, if we found *w* and *b* that make *C*(*w*,*b*) approaching to 0, then we in fact have found the right value of *w* and *b*. The job of learning algorithm is now to search for weights and biases that reduce the cost to minimum.

Before going further in a learning algorithm, it is better to stop a while on some of the ideas presented so far and match them to educational concepts. First, the *labeled training data* which contains the input values along with correct output assumes knowledge as something static and something we know in advance. Learning algorithm is not allowed to manipulate inputs or correct outputs in any case (Nielsen, 2015). This limits the ability of ANN to learn something previously known, not to discover something new. The idea of static knowledge contradicts with connectivism's principle of dynamic knowledge. The second important point here is how we could interpret this algorithm in an educational context. Let us continue with the connectivism's assumption that a single neuron represents a person. Thus, the inputs of the neuron would represent the training material or the current learning experiences. The correct outputs represent the reality (ontology) and neuron outputs represent a person's perceptions about the reality (epistemology). The difference between neuron outputs and correct outputs represents the gap between learner's perceptions and the reality. That is to say, learning is the process of minimizing the gap between learner's perceptions and the reality. Of course, this definition perfectly fits constructivist theory of learning. Jean Piaget (2013) interprets learning process mainly using two sub-processes: assimilation and accommodation. Assimilation refers to learner's tendency to use his current knowledge to deal with new situations while accommodation refers to learner's tendency to change his current knowledge when it conflicts with reality. This theory clearly matches the way used to teach a neuron using labeled training data. ANN researchers, furthermore, insist that, in the learning stage, one should look at the gap between epistemology and ontology not at the correctness of epistemology (Nielsen, 2015). The reason for this claim is that the number of correct answers is not smoothly related to the changes in weights and biases. That is, in learning stage, a teacher should not count how many times a learner gives correct answers and try to increase them. Instead, a teacher should focus on the gap between what a learner believes and the reality and try to decrease it. In other words, if a learner gives two wrong answers (123, 24) for a given question (5x5=?), these answers should not be treated equal. Because when a learner says 24, it seems he learned something closer to the reality even though the answer is not correct.

The time has come to see how a learning algorithm finds weight and bias that minimize the output of a cost function. To understand what the algorithm does, it is better to plot a cost function in relation to the variation of weight and bias. Since the output of a cost function (*C*) depends on weight (*w*) and bias (*b*) of the neuron, then we may plot *C* in three-dimensional space where each one of *w*, *b* and *C* represents one dimension.

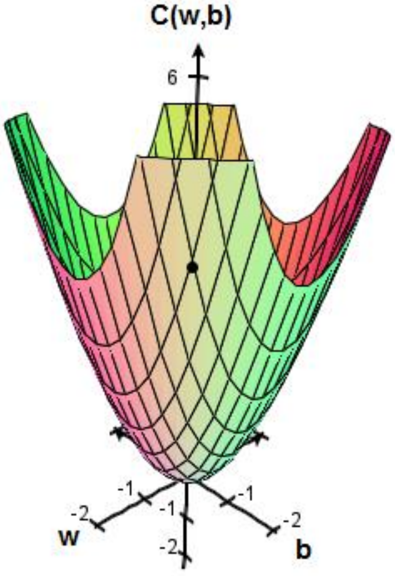

Fig. 13 Cost function in relation to weight and bias

The variation of weight and bias pair in relation to a cost function may constitute any terrain forms. Suppose it looks like a valley as shown in Fig. 13. Since we selected the value of weight and bias randomly at the beginning, the initial values of weight and bias can represent any point located on the surface. Suppose the point is located as shown. A learning algorithm should find a way to roll the point down the hill and make it settle at the bottom. Finding a right direction in three-dimensional space is not an easy task because it comprises watching the variation of three variables at once. It is better to split the task so we watch every two variables alone. To do so, we should pretend as if the third variable is constant.

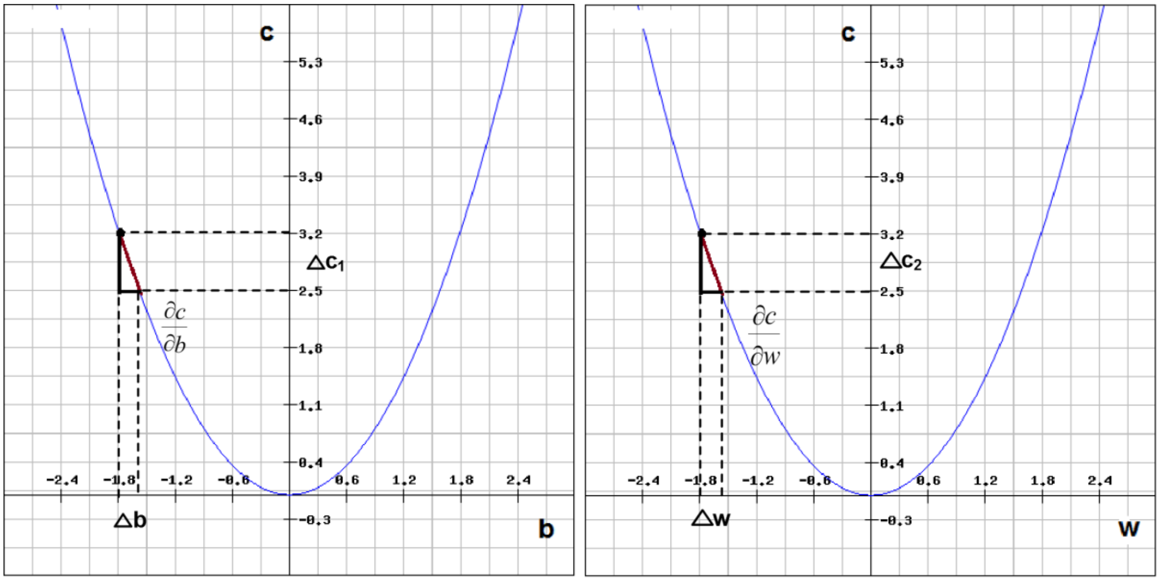

Fig. 14 (1) Cost function in relation to bias and (2) Cost function in relation to weight

In the first plot in Fig. 14, we are interested in finding a tiny change in bias that makes the cost smaller while keeping the weight constant (the same has to be said for the weight change in the second plot). To find this tiny change, we have to find a ratio (∂c/∂b) that relates the change of bias (Δb) to the change of cost while keeping the weight constant (ΔC_1). This ratio is known as partial derivative.

$$\Delta c_1 = \frac{\partial c}{\partial b}\Delta b \ , \ \Delta c_2 = \frac{\partial c}{\partial w}\Delta w \qquad (2)$$

A total change in cost is the summation of ΔC_1 and ΔC_2.

$$\Delta c = \frac{\partial c}{\partial b}\Delta b + \frac{\partial c}{\partial w}\Delta w \qquad (3)$$

Note that a change of cost ΔC also means the difference in the cost value before C_{old} and after C_{new} the change occurs.

$$\Delta c = c_{new} - c_{old} \qquad (4)$$

Since we need a new cost C_{new} to be smaller, this means ΔC should be negative. But how to guarantee that ΔC is always negative? If we choose Δw and Δb as following, this would guarantee ΔC to be always negative.

$$\Delta w = -\eta \frac{\partial \mathrm{c}}{\partial \mathrm{w}} \quad , \quad \Delta b = -\eta \frac{\partial \mathrm{c}}{\partial \mathrm{b}} \tag{5}$$

Why choosing these values of Δ*w* and Δ*b* would guarantee Δ*C* to be always negative? Because by substituting these choices into Δ*C* equation (3), one may easily find that these choices make sure Δ*C* negative.

$$\Delta c = -\eta \left[\left(\frac{\partial c}{\partial w} \right)^2 + \left(\frac{\partial c}{\partial b} \right)^2 \right] \tag{6}$$

Selecting Δ*w* and Δ*b* repeatedly as such will roll the point down the curve slowly and keep it settle at the bottom. This process is called gradient descent (see gradient in mathematics for general perspective and Nielsen, 2015 for specific discussion). So far so good but what we haven't mentioned yet what a factor (*η*) which appears in Δ*w*, Δ*b* and Δ*C* equations is called. They call it *learning rate*.

To understand why ANN researchers called *η* factor a learning rate, it is better to concentrate on one equation, take for example the equation of Δ*b*:

$$\Delta b = -\eta \frac{\partial \mathrm{c}}{\partial \mathrm{b}} \tag{7}$$

Note first that a sign of *η* refers to the direction in which we want to go to. A negative sign means we want to go down the curve. Now, if we choose *η* large, the step of Δ*b* becomes wide. And if we choose *η* small, the step of Δ*b* becomes tiny. Therefore, *η* controls the speed of Δ*b* learning. Since *η* appears on Δ*w* and Δ*C* equations as well, then we can say that *η* controls the speed of a neuron learning and it is logical to call it learning rate. It seems tempting to increase *η* so the step becomes wider and, hence, the point reaches the bottom faster. However, this is a false conclusion because a slope (∂*C*/∂*b*) is only valid for a tiny shift of *b*. To understand why, look at Fig. 15 below:

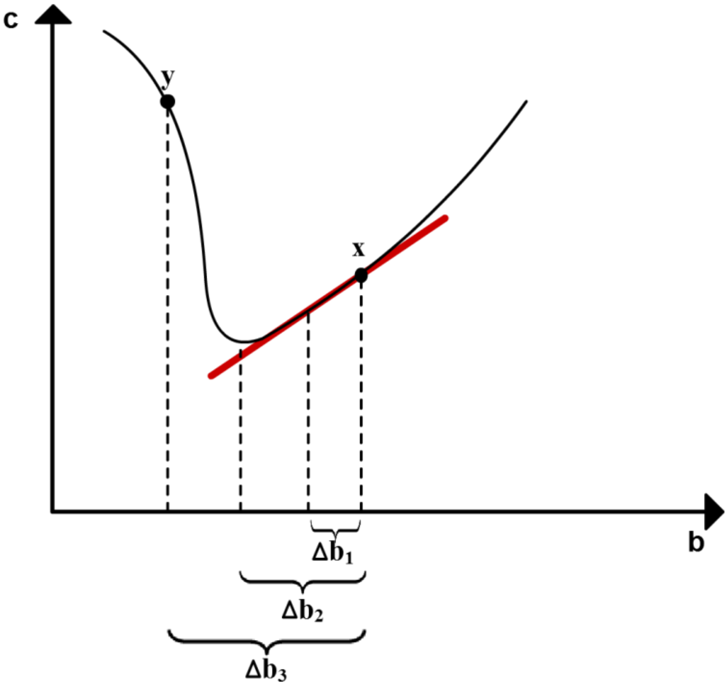

Fig. 15 Effect of selecting different value of bias on learning

Suppose you have a cost function in relation to bias as shown in the figure. The initial point is in (x). The red line represents the slope ($\partial C/\partial b$) at this point. First, if we choose η very small, the result would be a very small step like Δb_1. This is not a good strategy because it requires many steps before we reach the bottom of the curve. Therefore, learning becomes very slow. On the other hand, if we choose η very large, the result would be a very wide step like Δb_3. This is not a good strategy as well. It makes the point jump to higher cost (y). In this case, the right choice of η should be an intermediate value that produces step like Δb_2. How do ANN researchers tune the value of η? So far, there is no rule and they depend merely on try-and-error strategy (Nielsen, 2015)!

It is obvious that extracting an understandable educational interpretation out of this part of the algorithm is not an easy task. This paper does not also claim that it will offer a comprehensive interpretation. Instead, the interpretation coming shortly should be seen as an initial step toward understanding machine learning algorithms in a humanitarian learning context. Cumulative efforts from concerned researchers may eventually lead us to better understanding human and machine learning.

Recall that the cost represents a gap between learner's epistemology and ontology. Thus, one may argue that C_{old} represents the gap before passing through a learning experience. Likewise, C_{new} represents the gap after passing a learning experience. As a learner passes through a learning experience, the gap reduces from C_{old} to C_{new}. The gap shrinkage is ΔC. Therefore, ΔC represents the learning outcome. The learning outcome stems from the change in a bias of a learner's internal judgment system (Δb) and his own perspective (Δw). Changing student's perspectives Δw and his bias Δb toward smaller gap between epistemology and ontology ($C_{new} < C_{old}$) represents a learning outcome ΔC. Or in short, the learning outcome refers to the extent of progress a learner makes in bridging the gap between his epistemology and the ontology after passing learning experience.

A learning rate refers simply to the speed of learning outcome. Or how fast a learner should learn. The learning rate should not be too fast that makes a learner jump from point to point; long jumps disrupt learning. A learning rate should not be very slow too; it makes a learner crawl in details that would not serve him to achieve his goal. Finding the right pace of learning is a difficult task that depends on the initial state of the learner's perspectives and bias. The determinant factor of learner's speed of learning is where his epistemology is located in relation to the ontology. This may interpret why each learner has his own learning rate and why the same learner may change his rate from task to task.

One note in ANN model of learning is how AI researchers are setting the value of learning rate. Actually, learning rate is one of many other parameters which are left free for human and outside of ANN's control. For example, (1) the number of layers, (2) the number of neurons in each layer, (3) the size of training set, (4) the activation function type, and (5) regularization parameter as well as (6) the learning rate are some of those free parameters which are called hyper-parameters (Nielsen, 2015). Choosing the right values of hyper-parameters is left for a person who manages the ANN (see Fig. 16).

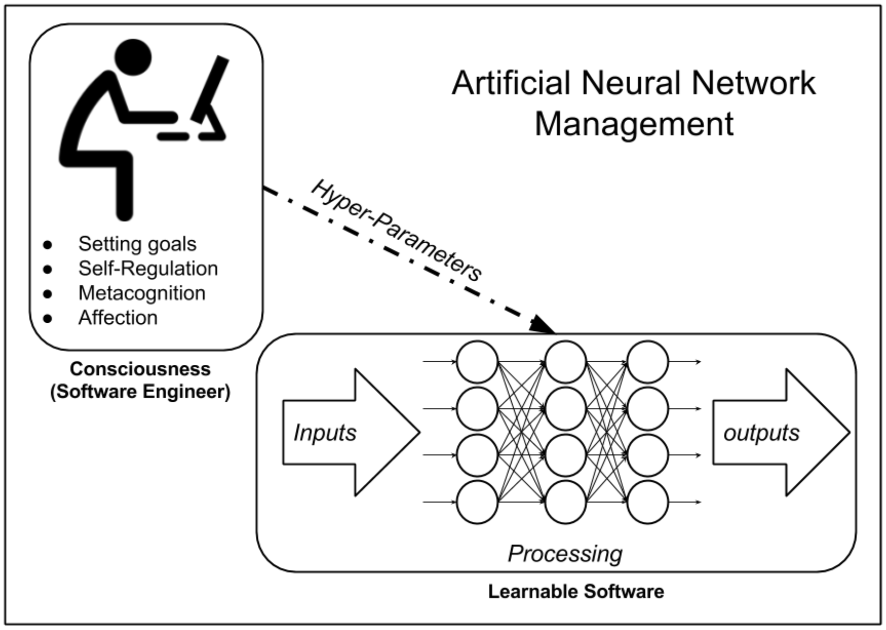

Fig. 16 Artificial Neural Network and human management

As we assumed in the beginning that ANN may represent a learner, this makes us wonder what does this person who is playing with hyper-parameters represent in a human's mind? Arguably, hyper-parameters are a way in which the learner exercises control over his thoughts and learning and, therefore, this person represents *human' agency* or *consciousness*. Bandura (2006) contends that human agency has four core properties which distinguish humans from automatons: intentionality, forethought, self-reactiveness, and self-reflectiveness. These four properties are sometime referred in educational literature as *self-regulation* and *metacognitive processes*. According to Vermunt and Verloop (1999), metacognitive regulation activities are those thinking activities learners use to decide on their goals, to exert control over their processing and affective activities and to steer the course and outcomes of their activities. For purposes of illustration, consider the following analogy of a software engineer and a learnable software. The software engineer represents the consciousness who sets the goals, plans for experiences, monitors and evaluates the progress of learning. A learnable software represents a neural pattern written in the brain. This learnable software is an elastic program which can automatically detects its mistakes and rewrites itself but under supervision of the engineer. The engineer manages, directs, and gives instructions to the learnable software but does not engage in writing the software by hand. The engineer is not a programmer and he does not even aware of how the software is written. Once the software is written 'correctly', the consciousness releases its control over the written software and the software is working deliberately. Only when something goes unexpected, the consciousness comes back to exercise control and regulates the process of rewriting the software again. Bandura (1999, 2006) criticizes connectionist's view of human learning as it concentrates merely on neural patterns to interpret learning and argues that this view strips humans of agentic capabilities and a self-identity. In contrary, Bandura (2006) conceives consciousness as an emergent property of brain activities which is not reducible solely to the property of neurons activity. In other words, the consciousness is higher-level force which is a result of lower-level neural activities but its properties are not limited to them. As clarified in this study, ANN design shows the need for consciousness force to manage and regulate ANN learning but this force does not occur as an emergent property of neural activity as Bandura proposes. Rather, it is a completely distinct entity which uses, guides and manages the neural activity and does not result from it. Siemens (2005) defines learning as a process of connecting specialized nodes but, as far as we know, connectivism does not refer to learning rate and other hyper-parameters in its assumptions. Connectivism is also criticized for its ambiguity in that it does not show how pattern recognition is done (Aldahdouh et al., 2015). It is not clear in connectivism what the characteristics of pattern recognition are.

Up to now, the paper presents learning in one neuron level. Even though the rules of learning in one neuron level are applicable for a network level, the complexity and time spent in training a network increase dramatically. Suppose that you have a deep fully-connected network as shown in Fig. 17:

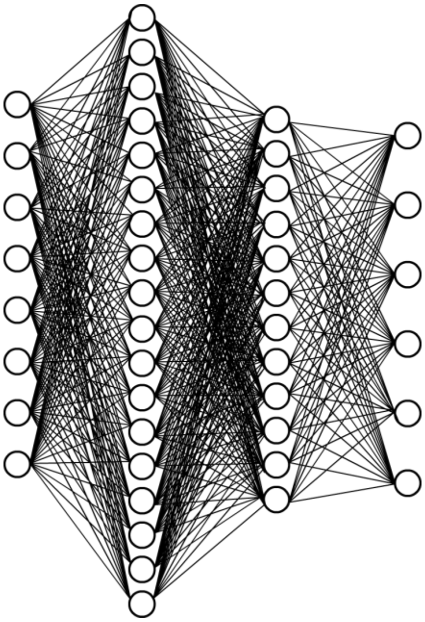

Fig. 17 Fully connected neural network with 44 neurons

In this network, you need to repeatedly calculate biases and weights according to the equations:

$$b_{new} - b_{old} = -\eta \frac{\partial \mathrm{c}}{\partial \mathrm{b}} \quad , \quad w_{new} - w_{old} = -\eta \frac{\partial \mathrm{c}}{\partial \mathrm{w}} \qquad (8)$$

Old values of weight (w_{old}) and bias (b_{old}) were given from the previous step or were set randomly at the first step. You provide an arbitrary value for learning rate (η) and tune it by try-and-error strategy. The remaining part is to calculate partial derivatives ($\partial c/\partial b$) and ($\partial c/\partial w$). Our network has 44 neurons but a typical ANN may have millions of neurons. In such networks, finding gradient becomes a tedious task. Part of the difficulty in finding partial derivatives for each neuron returns to the fact that a tiny shift in single neuron weight or bias will propagate to all neurons in the next layer. And the next layer will propagate changes to the next layer, and so on. This tiny shift significantly changes the cost of the whole network. ANN researchers have found a way to trace these changes called *back-propagation*. Back-propagation outperformed all other methods used previously to compute gradient (Goodfellow et al., 2016; Nielsen, 2015). The core idea of back-propagation depends on calculating partial derivatives using multivariable chain rule in mathematics. Presenting the mathematical model of back-propagation does not serve the purpose of this study. Intuitively speaking, back-propagation starts by computing the error in the output layer and then uses this error to compute the error in the preceding layers, one after another. That is, it goes through the network backward. At the end, it uses the error matrix to calculate the gradient that is required to compute the next values of bias b_{new} and weight w_{new} (see Goodfellow et al., 2016 and Nielsen, 2015 for more details).

Combined with back-propagation, gradient descent algorithm moves gradually to reduce the gap between correct output and network output. In each step of this movement, it computes the cost function and its gradient. The result is that each neuron in the network moves one step toward learning its *right* weights and bias. Eventually, the gradient descent reaches a point in which it can't reduce the cost anymore. At that point, the network reaches the maximum approximation to the correct output. This process is done automatically by gradient descent algorithm. In a network of millions of neurons, what are those *right* weights and bias that a single neuron learns? What is the meaning of those connections and biases? Why does each neuron connect to other neurons in that way? Until now no one has a theory. As Nielsen (2015) put it,

> In the early days of AI research people hoped that the effort to build an AI would also help us understand the principles behind intelligence and, maybe, the functioning of the human brain. But perhaps the outcome will be that we end up understanding neither the brain nor how artificial intelligence works (ch. 1)!

Learning in network level clearly supports the core assumption of connectivism: A single connection between two nodes does not have meaning in its own. A meaning is distributed across group of connections or patterns (Aldahdouh et al., 2015; Downes, 2006). Looking at the network from higher level mitigates its complexity (Downes, 2016). But that does not give us the answer and the exact meaning of the entities in the lower level. The inner entities are well organized and they certainly serve the purpose of the whole network but by looking at single entity or small number of entities, one may fall in the illusion of finding conflicting and contradictory ideas with the whole network.

Discussion

It has been argued that machine learning has nothing to do with human learning. Some researchers in education field do not even see the difference between recent technologies and traditional tools like books (Verhagen, 2006). Connectivists, on the other hand, argue that learning resides in non-human appliances (Siemens, 2005). They insist on the relevance of machine learning and, frequently, share ANN findings to support their assumptions. This paper tried to bridge the gap and examine the relationship between connectivism's assumptions and AI findings. Table 1 below summarizes, in points, the relationship between ANN concepts and connectivism's assumptions.

Table 1

Relationship between ANN concepts and connectivism's assumptions

ANN concept	Yes	No	Not clear	Clarification
Perceptron neuron		X		The connection is not necessarily sharp; it can be graded.
Neuron bias	X			A bias of neuron represents learner autonomy.
Sigmoid neuron			X	The graded feature of sigmoid is congruent with connectivism but the flat part of the curve is not.
Neuron as logic gate	X			A group of logic gates interprets the emergent property.

ANN concept	Yes	No	Not clear	Clarification
Network Layers		X		Arranging network in layers contradicts with chaos theory.
Feedforward net		X		Information in connectivism flows in both directions.
Recurrent net	X			Self-join connections are allowed in connectivism
Fully connected net	X			Connectivism seeks to increase connectivity as much as possible.
Convolutional net		X		Connectivism does not see the value of limiting connectivity.
Supervised learning		X		Supervised learning techniques are congruent with constructivism and former learning theories.
Unsupervised learning	X			Unknown and dynamic knowledge can only be learned by pattern recognition.
Learning rate			X	Connectivism has been criticized for its ambiguity in interpreting the process of pattern recognition.
Learning in network level	X			A single connection between two nodes does not have meaning in its own as connectivism suggests.

In this paper, we argue that the definition of learning as pattern recognition can only describe a learning process under certain conditions. As shown in the paper, ANN researchers use constructivist’s principles for teaching a network. Although not mentioned previously, we found that AI researchers use almost all previous learning theories. For example, one technique in AI mimics behaviorism and it is called *reinforcement learning*. In this technique, the agent learns from a series of *rewards* and *punishments* (Russell & Norvig, 2010). ANN researchers also use a technique called *Long-Short Term Memory* which mimics cognitivism (Olah, 2015). But all these methods should be seen from the lens of known and static knowledge only. That is, if the knowledge is known and static, ANN researchers may use labeled training data and make use of the gap between the network output and the correct output. In a situation where knowledge is not known or is not static, all these techniques will fail. The question of how to teach a network in these conditions remains not fully answered in AI field. As Russell and Norvig (2010) say, "how do we know that the hypothesis *h* is close to the target function *f* if we don't know what *f* is?" (p. 713). In such cases, the only hope is to use *unsupervised learning model* in which a learner should extract the pattern from given examples without explicit feedback. The repetition and relative similarity between objects in given examples may help a learner to cluster and combine different ideas together to come up with new object. And that is where connectivist's theory lies.

Even though connectivism’s assumptions are not congruent with some points in ANN, they are certainly valuable in interpreting machine learning algorithm in general. Assuming knowledge as a network; viewing learner as a node in network; and the connectivity between learners serve us to understand the relationship between neurons and the complexity of artificial learning network. Without such assumptions, one may find it extremely difficult to approach machine learning algorithm from an educational perspective.

Studying ANN reveals that the flow of information should be free but educators should not seek to increase the connectivity as much as they can. A fully-connected learning environment increases the complexity and makes the learning process harder. The learner should be free to share knowledge and to act within a *reasonable* number of valuable connections. In addition, studying ANN shows that learning is a slow and smooth process. This is in comparison to knowledge which is changing rapidly. Connectivism presents two solutions for this paradox: (1) to allow students to copy-and-paste and (2) to store their information outside themselves. Educators are also invited to find other innovative solutions.

This paper presents a very limited part of ANN literature. There are still some other topics that are valuable and deserve careful study from an educational perspective. Such topics include the number of nodes in each layer; different types of neurons; different cost functions; weight initialization; overfitting and regularization; Long Short Term Memory technique; bidirectional recurrent network; and Deep Belief Network.

References

Aldahdouh, A. A., & Osório, A. J. (2016). Planning to Design MOOC? Think First! *The Online Journal of Distance Education and E-Learning*, *4*(2), 47–57. Retreived (https://www.tojdel.net/journals/tojdel/articles/v04i02/v04i02-06.pdf)

Aldahdouh, A. A., Osório, A. J., & Caires, S. (2015). Understanding knowledge network, learning and connectivism. *International Journal of Instructional Technology and Distance Learning*, *12*(10), 3–21. Retreived (http://www.itdl.org/Journal/Oct_15/Oct15.pdf#page=7)

Bandura, A. (1999). Social Cognitive Theory: An Agentic Perspective. *Asian Journal of Social Psychology*, *2*, 21–41. http://doi.org/10.1111/1467-839X.00024

Bandura, A. (2006). Toward a Psychology of Human Agency. *Perspective on Psychological Science*, *1*(2), 164–180. http://doi.org/150.131.73.116

Bell, F. (2010). Network theories for technology-enabled learning and social change: Connectivism and actor network theory. In *7th International Conference on Networked Learning* (pp. 526–533). Retrieved from http://usir.salford.ac.uk/9270/1/Bell.pdf

Bell, F. (2011). Connectivism: Its place in theory-informed research and innovation in technology-enabled learning. *The International Review of Research in Open and Distance Learning*, *12*(3), 98–118. http://doi.org/10.19173/irrodl.v12i3.902

Brabazon, T. (2016). *The University of Google: Education in the (post) information age*. Routledge. Retrieved from https://books.google.fi/books?hl=en&lr=&id=QdeXCwAAQBAJ&oi=fnd&pg=PT7&ots=JbNthv0zbm&sig=QTtI95zOhlT9ibCjonqJeuO3he8&redir_esc=y#v=onepage&q&f=false

Britz, D. (2015). Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs. Retrieved August 22, 2016, from http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

Clarà, M., & Barberà, E. (2014). Three problems with the connectivist conception of learning. *Journal of Computer Assisted Learning*, (OCTOBER 2013). http://doi.org/10.1111/jcal.12040

Cormier, D. (2016). Content is a print concept. Retrieved June 19, 2016, from http://davecormier.com/edblog/2016/06/11/content-is-a-print-concept/

Dirckinck-Holmfield, L., Jones, C., & Lindström, B. (2009). *Analysing networked learning practices in higher education and continuing professional development*. Sense Publishers.

Downes, S. (2005). An Introduction to Connective Knowledge. In K. Habitzel, T. D. Märk, & S. Prock (Eds.), (pp. 77–102). innsbruck university press. Retrieved from http://www.oapen.org/download?type=document&docid=449459#page=84

Downes, S. (2006). Learning Networks and Connective Knowledge. In H. H. Yang & S. C.-Y. Yuen (Eds.), *Collective intelligence and elearning* (pp. 1–26). New York: Information Science Reference. Retrieved from https://books.google.fi/books?hl=en&lr=&id=GGx0GlvbYa0C&oi=fnd&pg=PA1&dq=Learning+Networks+and+Connective+Knowledge&ots=t48QLHJ56R&sig=z-9LiYdqRJG8H4Syp2fkT9WW5Z4&redir_esc=y#v=onepage&q=Learning Networks and Connective Knowledge&f=false

Downes, S. (2009). New Tools for Personal Learning. Retrieved May 11, 2016, from http://www.downes.ca/presentation/234

Downes, S. (2010a). Dimensions of a Learning Network. Retrieved January 10, 2017, from http://www.slideshare.net/Downes/dimensions-of-a-learning-network-5560762

Downes, S. (2010b). What is democracy in education. Retrieved February 1, 2016, from http://halfanhour.blogspot.pt/2010/10/what-is-democracy-in-education.html

Downes, S. (2012). *Connectivism and Connective Knowledge: essays on meaning and learning networks*. *Stephen Downes Web*. Retrieved from http://www.downes.ca/files/books/Connective_Knowledge-19May2012.pdf

Downes, S. (2016). NRC01PL 07 - The MOOC Ecosystem - YouTube. Retrieved September 2, 2016, from https://www.youtube.com/watch?v=kxstCBbIrUs&feature=youtu.be

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Book in preparation for MIT Press. Retrieved from http://www.deeplearningbook.org/

Kop, R., & Hill, A. (2008). Connectivism: Learning theory of the future or vestige of the past? *International Review of Research in Open and Distance Learning*, *9*(3). http://doi.org/10.19173/irrodl.v9i3.523

Nielsen, M. a. (2015). *Neural Networks and Deep Learning*. *Determination Press*. Determination Press. Retrieved from http://neuralnetworksanddeeplearning.com/

Olah, C. (2014). Conv Nets: A Modular Perspective. Retrieved August 23, 2016, from http://colah.github.io/posts/2014-07-Conv-Nets-Modular/

Olah, C. (2015). Understanding LSTM Networks. Retrieved August 22, 2016, from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Piaget, J. (2013). *The construction of reality in the child* (Vol. 82). Routledge. Retrieved from https://books.google.fi/books?hl=en&lr=&id=PpfGxMDZP-4C&oi=fnd&pg=PP2&dq=The+construction+of+reality+in+the+child&ots=LLsJNq8Qud&sig=RtAqeUgF2hA30rsG9-GOW4e9kn4&redir_esc=y#v=onepage&q=The construction of reality in the child&f=false

Russell, S., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach* (Third Edit). New Jersey: Prentice Hall. Retrieved from http://aima.cs.berkeley.edu/

Siemens, G. (2005). Connectivism: A learning theory for the digital age. *International Journal of Instructional Technology and Distance Learning*, *2*(1), 3–10. Retrieved from http://www.itdl.org/Journal/Jan_05/article01.htm

Siemens, G. (2006). *Knowing knowledge*. Lulu. com. Retrieved from http://www.elearnspace.org/KnowingKnowledge_LowRes.pdf

Siemens, G., & Tittenberger, P. (2009). *Handbook of emerging technologies for learning*. Manitoba, Canada: University of Manitoba. Retrieved from http://elearnspace.org/Articles/HETL.pdf

Verhagen, P. (2006). Connectivism : a new learning theory? *University of Twente*. Retrieved from http://elearning.surf.nl/e-learning/english/3793

Vermunt, J. D., & Verloop, N. (1999). Congruence and friction between learning and teaching. *Learning and Instruction*, *9*(3), 257–280. http://doi.org/10.1016/S0959-4752(98)00028-0

Vermunt, J. D., & Vermetten, Y. (2004). Patterns in student learing: relationships between learning strategies, conceptions of learning and learning orientations. *Educational Psychology Review*, *16*(4), 359–384. http://doi.org/10.1007/s10648-004-0005-y

Wang, Z., Chen, L., & Anderson, T. (2014). A Framework for Interaction and Cognitive Engagement in Connectivist Learning Contexts. *International Review of Research in Open and Distance Learning*, *15*(2), 121–141. http://doi.org/10.19173/irrodl.v15i2.1709

About the authors

Alaa A. AlDahdouh PhD candidate, Institute of Education of University of Minho, Portugal. He has B.S in computer engineering with cumulative experience in software programming. He is interested in studying human and machine learning as well. His work focuses on investigating new educational theories such as connectivism and their role to enhance educators' understanding of learning process.
E-mail: AlaaAldahdouh@gmail.com
Find him on Social Media: ResearchGate & Twitter