Basic Overview of Machine Learning Methods

JEFF GILL

Distinguished Professor
Department of Government, Department of Mathematics & Statistics
Member, Center for Neuroscience and Behavior
Founding Director, Center for Data Science
American University

Introduction to Neural Networks

▶ NN are a "black box" machine learning to explain some set of outcomes given a set of inputs.

▶ Also a classification problem.

▶ This means that the coefficients and values on the internal layers are not interpretable, and the only result with any value is the final result.

▶ The network analysis is neurological/cerebrial wherein learning is done through a serial training processes, which can be quite complex computationally.

▶ Unlike more conventional social science regression-style inference the (almost) only concern is explaining the outcome regardless of complexity or generalizability within.

▶ The most general way to think about NNs is as a two-stage regression/classification model represented by layers in a network diagram.
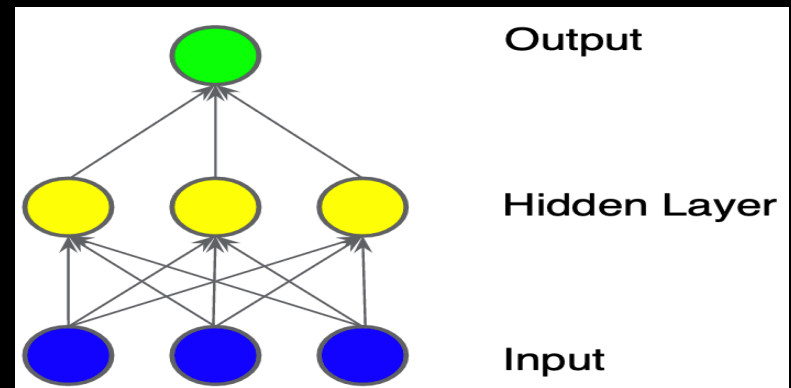
Introduction to Neural Networks

▶ The starting point is a representation of a simple *linear* model as a graph.

▶ The blue circles are input features.

▶ The green circle is the *weighted* sum of the inputs, which is the output (classification) of interest).

▶ This looks like a causal diagram, but it is not meant to imply causality.
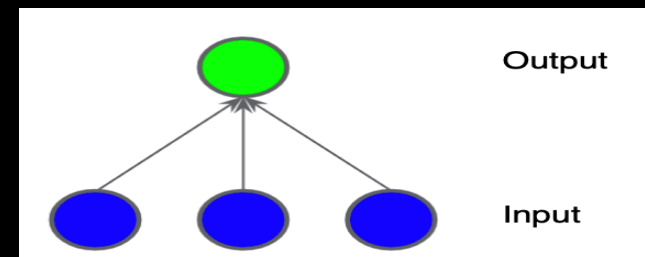
Introduction to Neural Networks

▶ Now consider a single *hidden layer* of intermediate values from an intermediate weighting process that is conditional on each input features as a weighted sum of these.

▶ This is the yellow circles in the figure.

▶ The weights could be (and many are in complex models) equal to zero in the first level relationships.

▶ Now the output is a direct weighted linear sum of the hidden layer nodes.

Introduction to Neural Networks

▶ A single hidden layer is unusually simple for real problems, so we can have multiple iterations, each conditional on the previous hidden layer with weighting.

▶ This is still a linear construction involving matrix algebra.
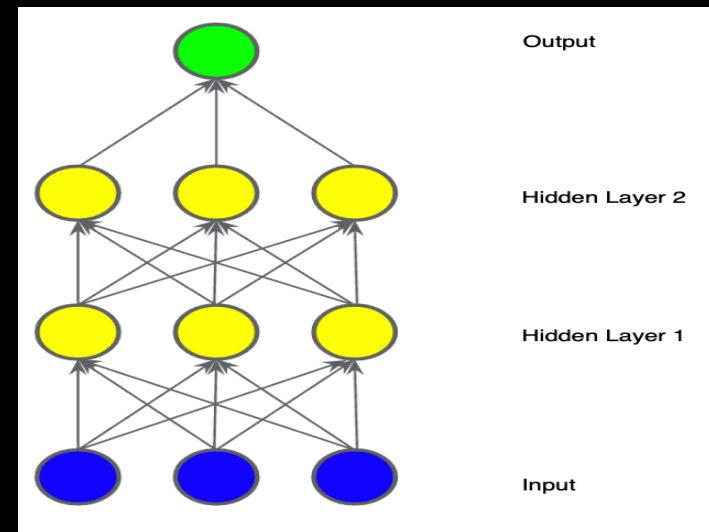
Introduction to Neural Networks

▶ Of course we want to represent/solve nonlinear relationships and this is done by piping the linear weighted relationships through a nonlinear function called an *activation function*.

▶ In the figure there is a single layer of activation functions between two hidden layers.

▶ In practice we can model very complicated relationships between the inputs and the final output by stacking many linear and nonlinear layers inbetween, creating higher level functionality.
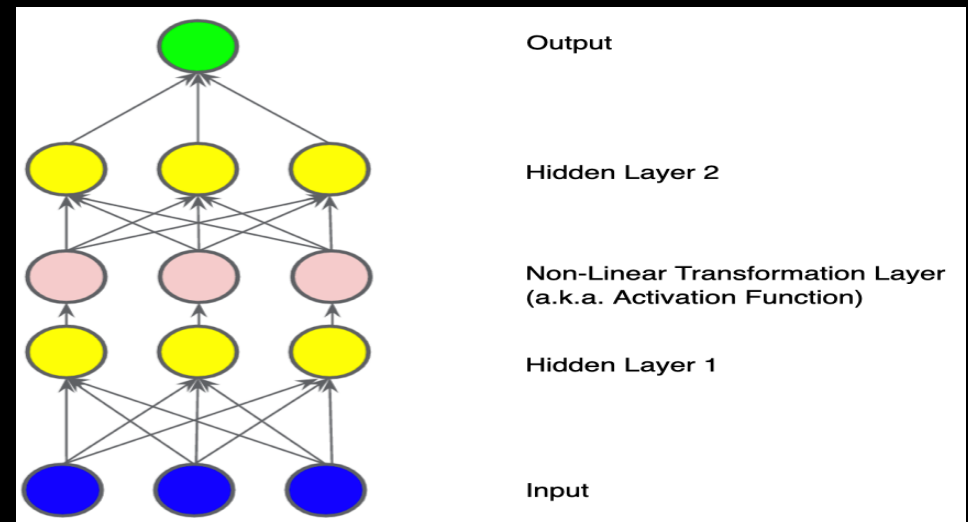
Specific Example of a Neural Network
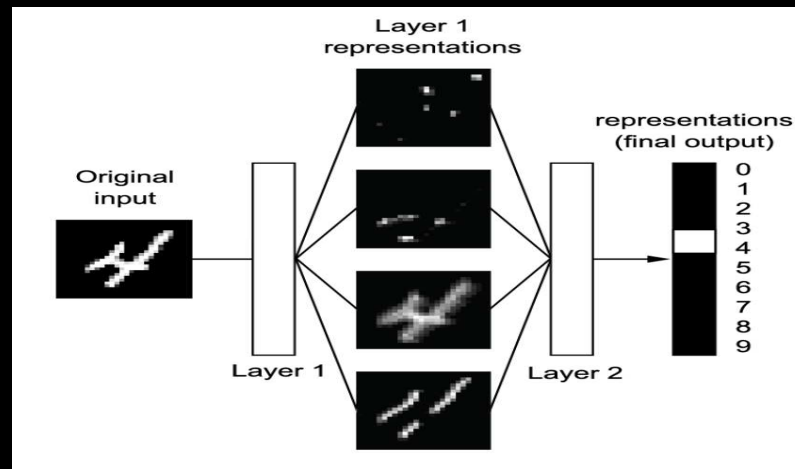
From: Deep Learning with Python, Second Edition by François Chollet, Manning.

Neural Networks Development

▶ Phases: training, vesting, validating.

▶ The training data must have outcome labels: $\mathbf{y}$.

▶ The testing data hides the label and is used for prediction and comparison to training data labels: $\mathbf{y} - \hat{\mathbf{y}}$.

▶ There are many forms of assessment for neural networks but they basically compare known outcomes to predicted outcomes.

Technical Basis of Neural Networks, Terms

▶ We start with an input matrix of information denoted $\mathbf{X}$.

▶ Consider a K-class outcome at the "top" of the diagram where the kth unit or category modeling the probability of this as an outcome.

▶ So the outcomes (targets) are denoted $Y_k, k = 1, 2, \ldots, K$, each denoted as a zero or one for a given target.

▶ The *derived features* in a hidden layer are created from weighted linear combinations of their inputs.

▶ Suppose we have a single hidden layer with $m = 1, 2, \ldots, M$ derived features denoted Z_m.

▶ Consider also an *activation function*, σ that probabilistically or deterministically decides a connection is "turned on."

Technical Basis of Neural Network, Steps

▶ For a single X in $\mathbf{X}$ the target vector Y_k are modeled as a linear combination of the Z_m according to three steps.

▶ STEP 1: the derived features in the hidden layer are produced by a fitted linear regression model called the activation function:

$$Z_m = \sigma(\alpha_{0m} + X\alpha_m)$$

where the "constant" term is called the bias unit or bias neuron and is more important in smaller models (characteristics with a value of 1 that the neural network did not previously have).

▶ STEP 2: the linear outcome component is produced by another fitted linear regression model:

$$T_k = \beta_{0k}Z_m\beta_k$$

▶ STEP 3: to be as general as possible, allow a final transformation of the vector of outputs T:

$$f(\mathbf{X}) = g_k(T_k)$$

Technical Basis of Neural Network, Details

▶ For regression the $g_k(T_k)$ function can be just the identity, but the softmax function is preferred because it provides estimates that sum to one:

$$g_k(T_k) = \frac{\exp(T_k)}{\sum_{\ell=1}^{K} \exp(T_\ell)}$$

(notice that this is really logit).

▶ The Z_k in the middle are referred to as hidden units because they are not directly observable.

▶ The *activation function* (σ) is apply named since they determine whether there will be an edge (connection) between a given X and a given Z.

▶ If σ is just the identity function then STEP 1 and STEP 2 just collapse to a simple linear model in the inputs.

Activation Functions

▶ There are two very common forms which are really like CDFs.

▶ *Sigmoid* (logit), $F(x) = [1 + \exp(-x)]^{-1}$.

▶ *Rectified Linear Unit* (ReLu), $F(x) = \max(0, x)$.

▶ As with anything in this realm there are many alternatives in use.
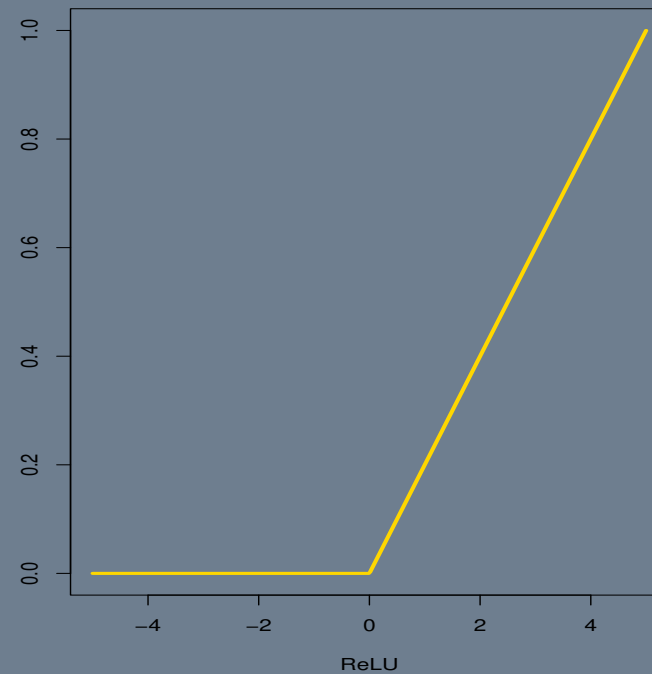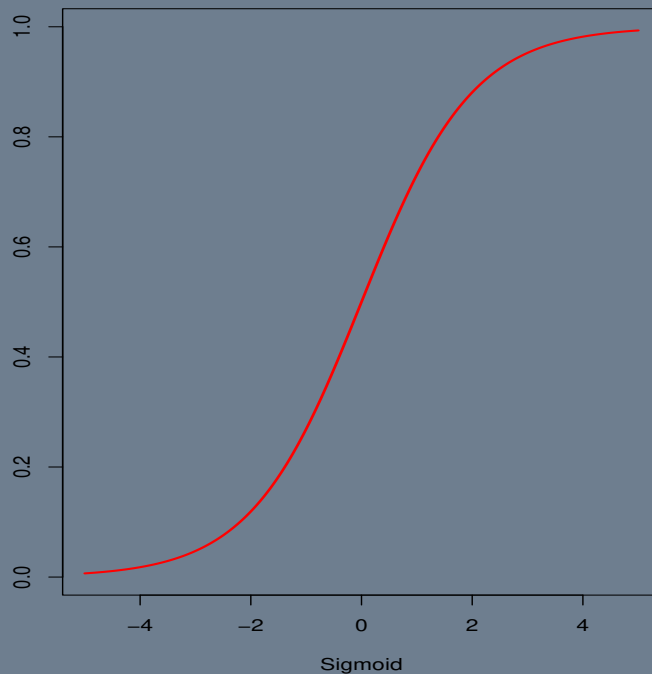
Deep Learning Illustration
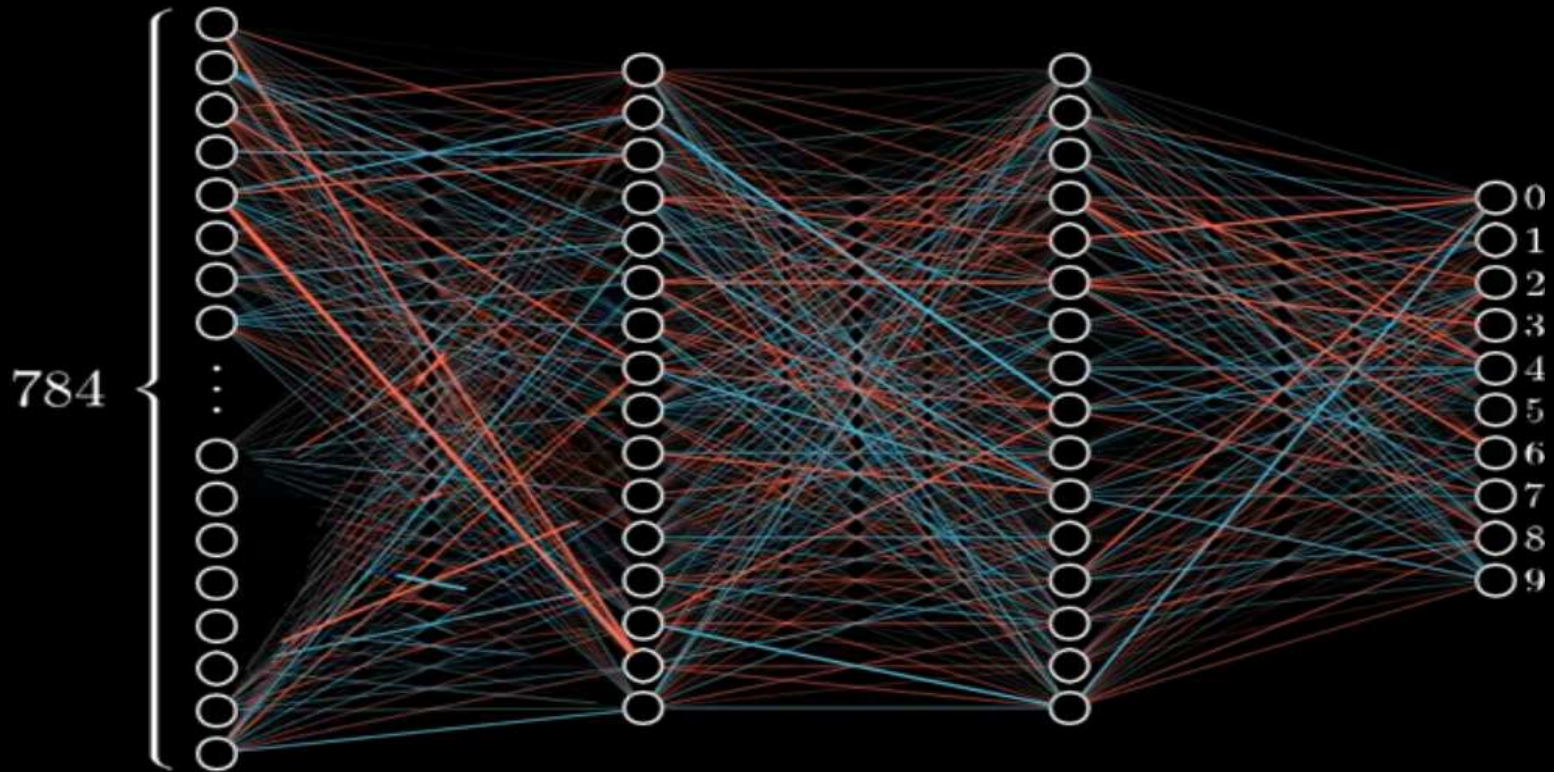
From Chollet, Francois. *Deep learning with Python*. Simon and Schuster, 2021.

Neural Networks and Deep Learning

▶ For a long time researchers struggled to find a way to train neural nets, without success. Then in 1986, David Rumelhart, Geoffrey Hinton and Ronald Williams published the paper: "Learning Internal Representations by Error Propagation" introducing the backpropagation training algorithm, which is still the standard today.

▶ Their idea is based on Gradient Descent using an efficient technique for computing the gradients automatically.

▶ In two passes through the network (one forward, one backward), the backpropagation algorithm computes the gradient of the network's error with regards to every single model parameter.

▶ It finds out how each connection weight and each bias term should be tweaked in order to minimize the error.

▶ With these gradients, it performs a regular Gradient Descent step, and the whole process is repeated until the network converges to the solution.

Deep Learning Algorithmic Forward Pass Details

▶ Process mini-batch one at a time going through the full training set multiples where each pass is called an epoch.

▶ These mini-batches are then passed to the input layer of the network, which then sends it to the first hidden layer.

▶ The initial weights of the hidden layers are randomly assigned.

▶ Then compute the output of all the neurons in this layer for every mini-batch instance.

▶ This result is then passed to the first hidden layer.

▶ Then repeat this process at this next layer, and so on until the last layer is reached: the output layer.

▶ All of these intermediate results are recorded for the backward pass.

▶ This completes the forward pass.

▶ Now calculate the network error: $\mathbf{y} - \hat{\mathbf{y}}$.

The Chain Rule

▶ This provides a means of differentiating nested functions of the form $f \circ g = f(g(x))$.

▶ The case of $g(x)^{-1} = (4x^3 - 2x)^{-1}$ fits this categorization because the inner function is $g(x) = 4x^3 - 2x$ and the outer function is $f(u) = u^{-1}$.

▶ Usually u is used as a placeholder here to make the point that there is a distinct subfunction.

▶ To correctly differentiate such a nested function, we have to account for the actual *order* of the nesting relationship, done by:

$$\frac{d}{dx}f(g(x)) = f'(g(x))g'(x),$$

provided that $f(x)$ and $g(x)$ are both differentiable functions.

▶ We can also express this in the other standard notation: if $y = f(u)$ and $u = g(x)$ are both differentiable functions, then

$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx},$$

which may better show the point of the operation.

▶ If we think about this in purely fractional terms, it is clear that du cancels out of the right-hand side, making the equality obvious.

The Chain Rule

▶ Let us use this new tool to calculate the function $g(x)^{-1}$ from above ($g(x) = 4x^3 - 2x$):

$$\frac{d}{dx}g(x)^{-1} = (-1)(4x^3 - 2x)^{-2} \times \frac{d}{dx}(4x^3 - 2x)$$

$$= \frac{-(12x^2 - 2)}{(4x^3 - 2x)^2}$$

$$= \frac{-6x^2 + 1}{8x^6 - 8x^4 + 2x^2}.$$

Deep Learning Algorithmic Backward Pass Details

▶ Starting with $\mathbf{y} - \hat{\mathbf{y}}$ for the smaller number of nodes in the output layer measure these error contributions backward for each connection in the layer below using the chain running backward until the input layer is reached.

▶ The reverse pass efficiently measures the error gradients across every connection weight in the layer below by propogating the error gradient backward through the networ..

▶ Thus the last step performs a Gradient Descent step to adjust all of the connection weights in the network using all of the individual error gradients just calculated.

▶ Now the Deep Learning network is fully trained and can be applied to different test data.

Gradient Descent for Machine Learning

▶ A gradient is a derivative that measures how much an (output) function changes for small changes in an input.

▶ In the deep learning sense it measure the change in all the weights for a small change in an error.

▶ Since it is a derivative, higher values mean that the slope is steeper and lower values mean that the slope is flatter.

▶ Higher slopes mean that the model can learn faster and zero slopes mean that no learning takes place.

▶ We want to get to the minimum (of errors) in a multidimensional sense, which is to minimize the function $J(w, b)$, where w is a set of weights, then a Gradient Step looks like:

$$b = a - \gamma \nabla f(a),$$

where b is the next location, a is the current location, γ is the learning rate or step size, and $\nabla f(a)$ is the direction of the step.
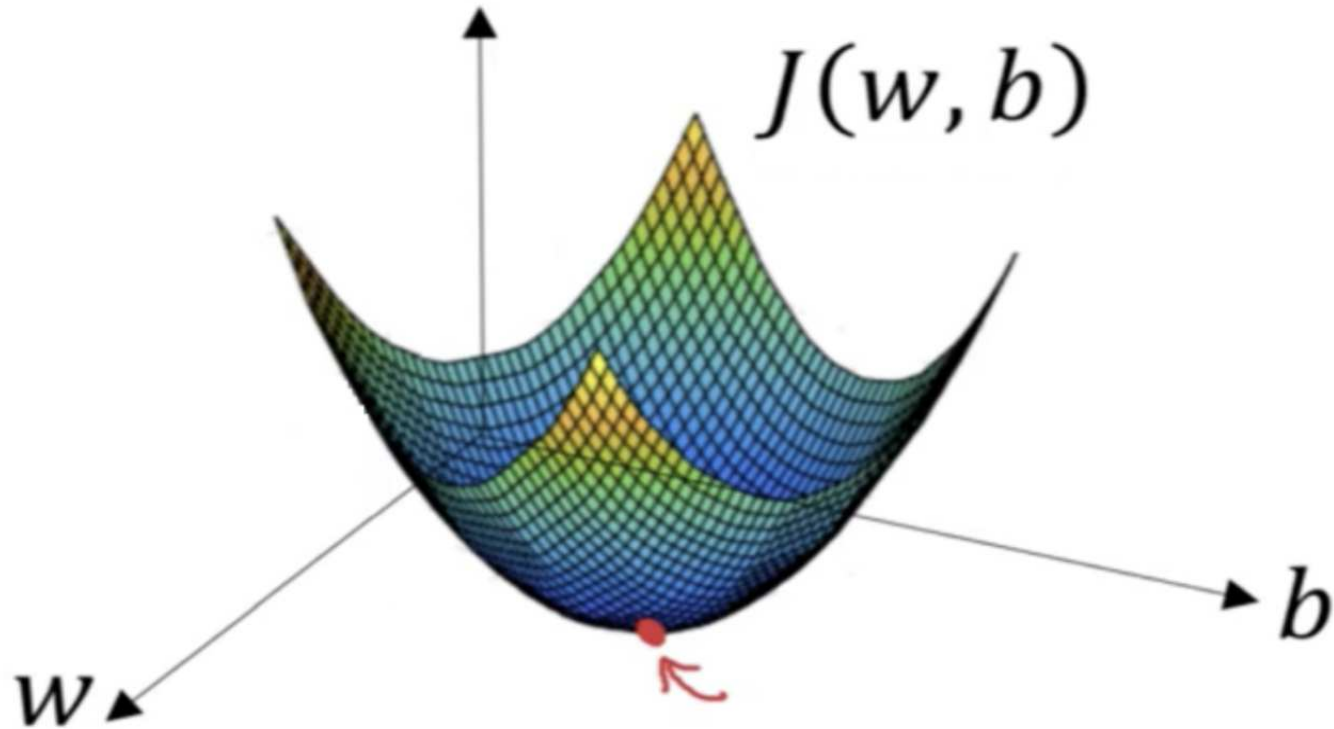
Gradient Descent Illustration

Image: Niklas Donges

Learning Rate

▶ This is the size of the steps that are taken to reach the function minimum.

▶ It is usually a small value that is evaluated and updated based on the behavior of the cost func- tion.

▶ A large value results in larger steps but risk overshooting the minimum.

▶ A small value is more precise, but lowers the efficiency since it takes more cycles to reach the minimum.
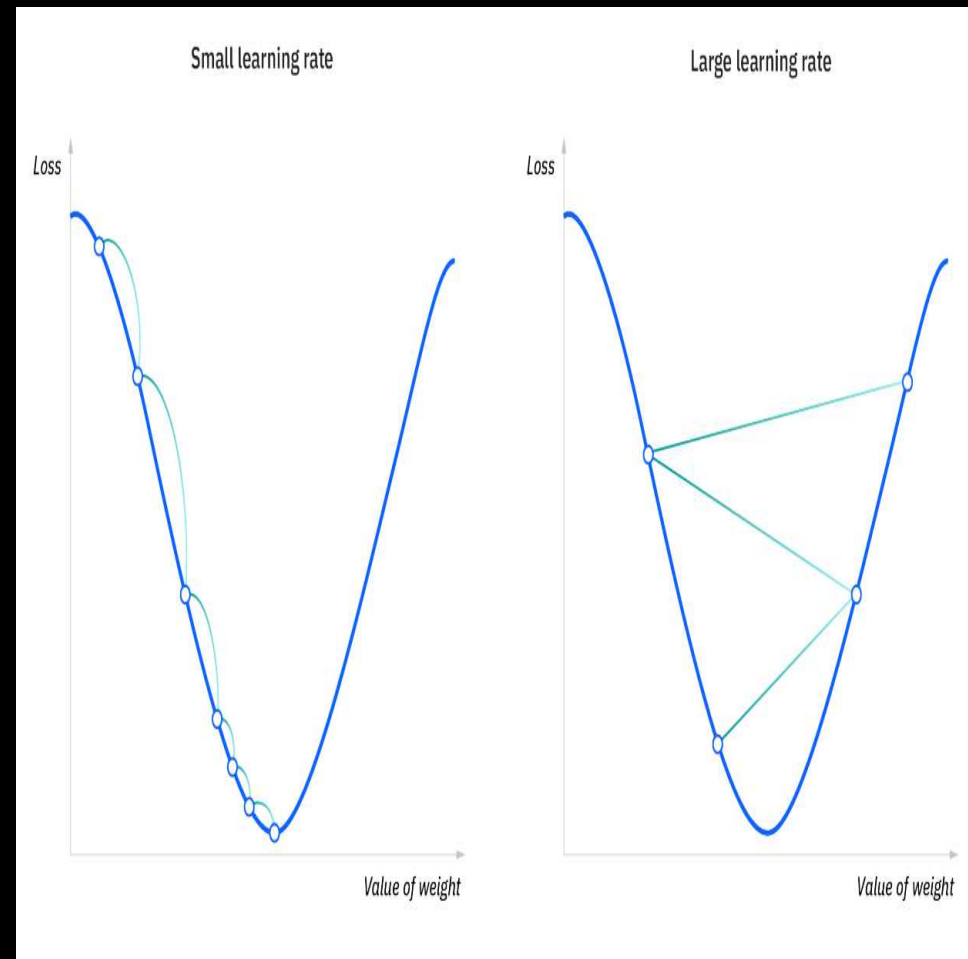
Source: IBM.

A Recent Development to Pay a Lot of Attention To

▶ "Chat Generative Pre-trained Transformer"

▶ ChatGPT is a breakthrough application of Deep Learning by OpenAI (then purchased by Microsoft) that can write essays, solve complex problems, seamlessly interact with humans, perform search in a way not seen before, and more

▶ Only in general use since November 2022

▶ As of December, ChatGPT had an estimated more than 100 million monthly active users

▶ ChatGPT's most recent GPT-3.5 model was trained on 570GB of text data from the internet, which OpenAI says included books, articles, websites, and even social media

▶ This is only the beginning of a revolution in AI power that is extremely easy to use

A Recent Development to Pay a Lot of Attention To

▶ The news media has recently exploded their coverage of OpenAI's partial release of ChatGPT

▶ "We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests"

▶ Representative Ted Lieu, Democrat of California, wrote in a guest essay in The New York Times in January that he was "freaked out" by the ability of the ChatGPT chatbot to mimic human writers

▶ Another Democrat, Representative Jake Auchincloss of Massachusetts, gave a one-minute speech written by ChatGPT calling for regulation of AI

▶ The problem is that most lawmakers do not even know what AI is, said Representative Jay Obernolte, a California Republican and the only member of Congress with a master's degree in artificial intelligence

A Recent Development to Pay a Lot of Attention To