Introduction of Neural Networks and Deep Learning

IBM Cloud and Cognitive Software

Topics for this session: Basic Neural Network

1. Neural network overview.
2. Different variants of Neural Networks and use cases
3. Top Deep Learning Frameworks
4. More on basic components of NN:

- Cost function, gradient descent, forward propagation and backward propagation, initialization, activation function,etc

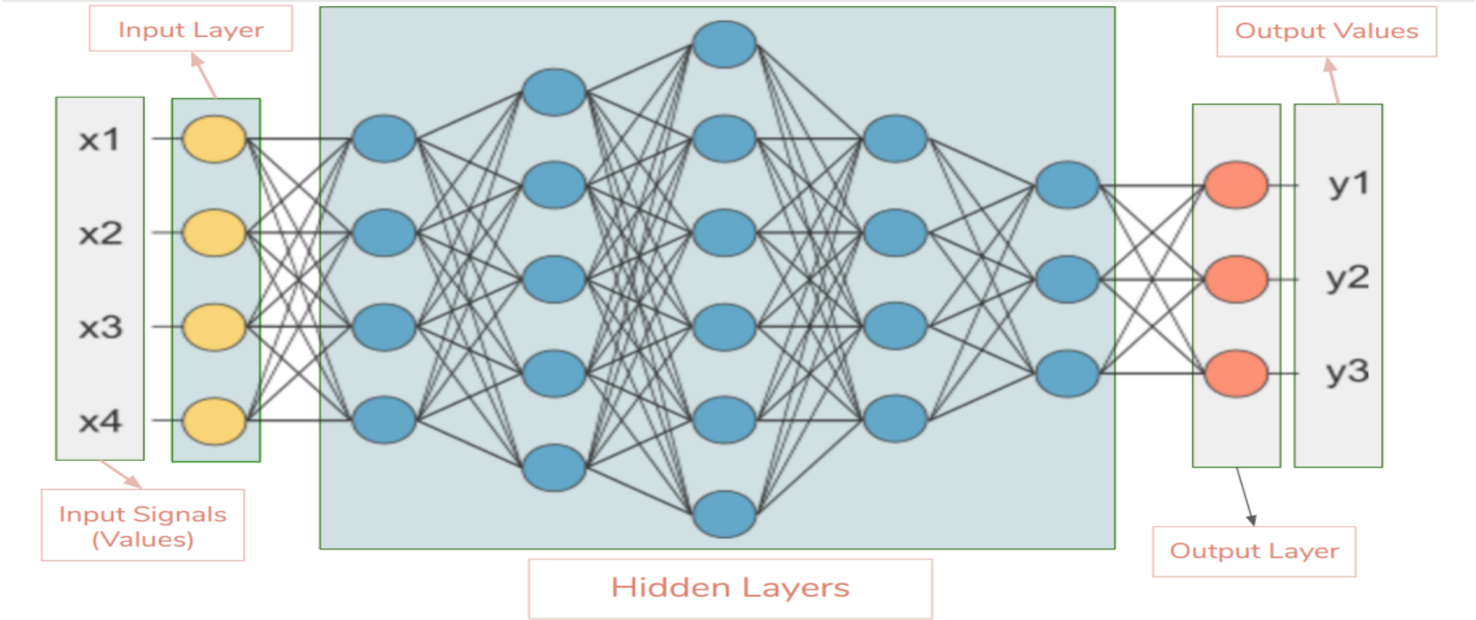
What are Neural Networks and Deep Learning

- "Neural network is a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs."

  -by Dr. Robert Hecht-Nielsen, the inventor of one of the first neurocomputers

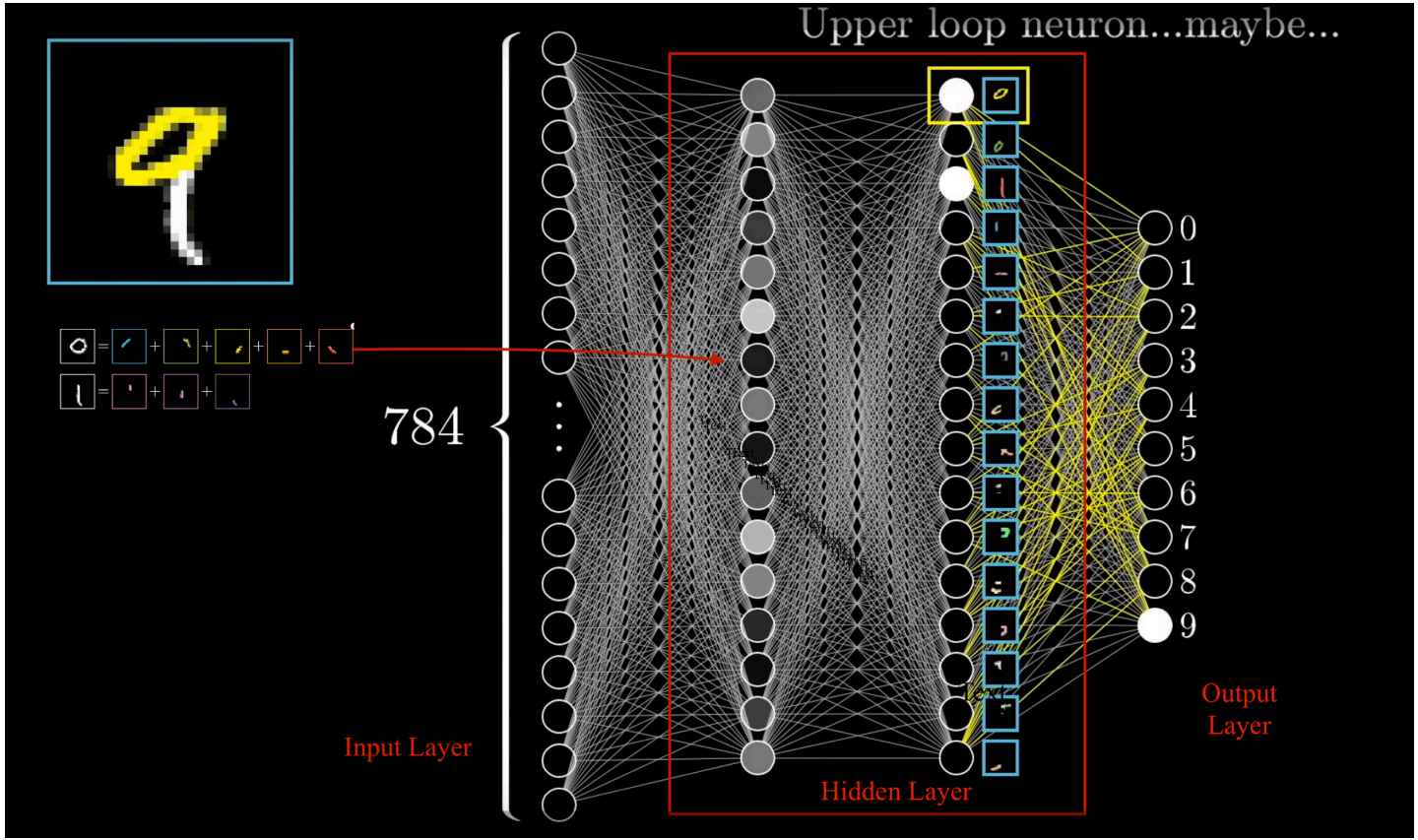
- Deep Learning, refers to training Neural Networks,sometimes very large Neural Networks.
- Industry applications: Google Lens, Google's self-driving car division, IBM Watson products etc.

Basic Neural Network Layout



Basic neural network layout

How does NN work?



Neural Networks (NN)

Basic questions:

- What kind of problem is NN good for?
  - Large input feature numbers (n) and large predictor numbers
  - Availability of massive amounts of data
  - Complex structure in data
- NN could solve simple questions too but there are many other Machine learning methods to use already.
- Disadvantages might need to think:
  - complexity, resource-intensiveness and lack of interpretability
  - Why deep learning began to take off?

Different Variants of Neural Networks and Use Cases

Multilayer Perceptrons (MLPs)	Tabular datasets classification and regression
Convolutional Neural Networks (CNNs)	image recognition and classification, face detection, drug discovery and image processing
Recurrent Neural Networks (RNNs)	language translation, natural language processing entity extraction, speech analysis,music generation, handwriting generation

Basic Components in Neural Network

Basic Components in Neural Network

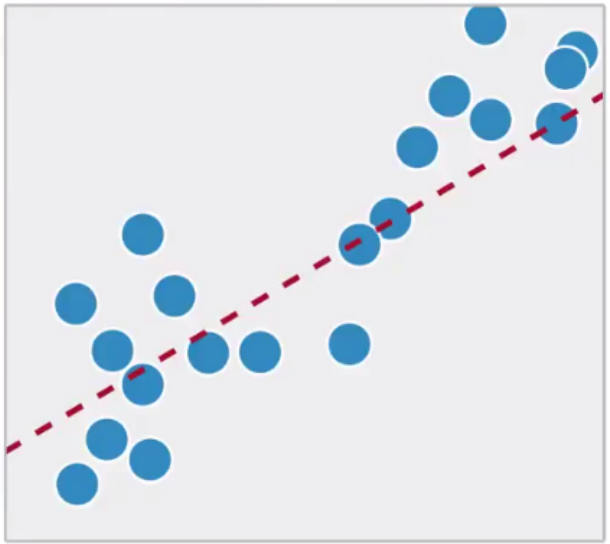
1. Cost function
2. Optimization using gradient descent
3. Activation function
4. Improving deep neural networks
5. Commonly asked questions

What is regression?

Regression is the process of predicting continuous values.

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

Continuous Values



Cost Function

$$\mathcal{J} = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}^{(i)}$$

for x in dataset:

$$\hat{y} = model_W(x) \qquad \text{(predict)}$$

$$W = W - \alpha\frac{\partial \mathcal{J}(y,\hat{y})}{\partial W} \qquad \text{(update parameters)}$$

Where:

- $\hat{y}$ is the model's prediction given an input x.
- W denotes the parameters.
- $\frac{\partial \mathcal{J}}{\partial W}$ is a gradient indicating the direction to push the value W to decrease $\mathcal{J}$.
- α is the learning rate which you can tune to decide how much you want to adjust the value of W per iteration.

Understanding gradient descent

Intuition for Gradient Descent

- an optimization algorithm used to find the values of parameters that minimizes a cost function.

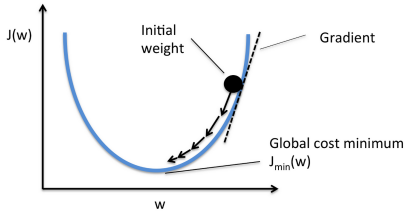
Batch Gradient Descent

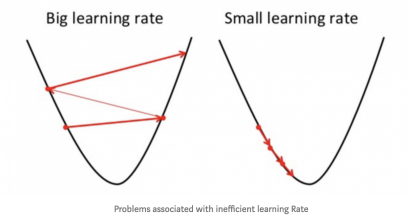
- The cost is calculated for a machine learning algorithm over the certain partition of training dataset for each iteration of the gradient descent algorithm.

Stochastic Gradient Descent

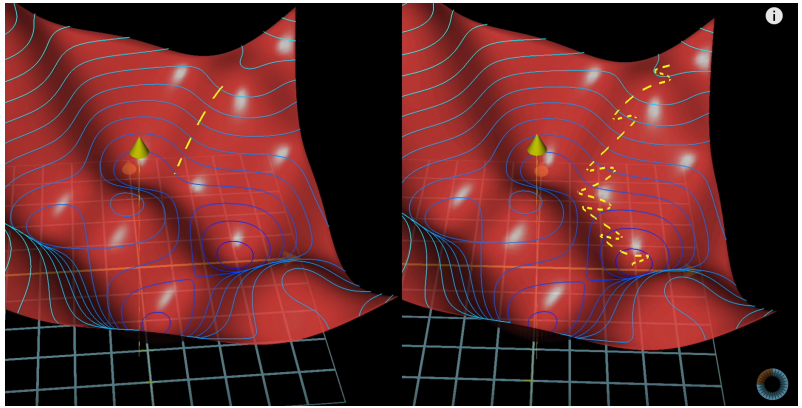
For very large dataset, computes the gradient using a

single sample,needs to slowly reduce the value of learning rate.



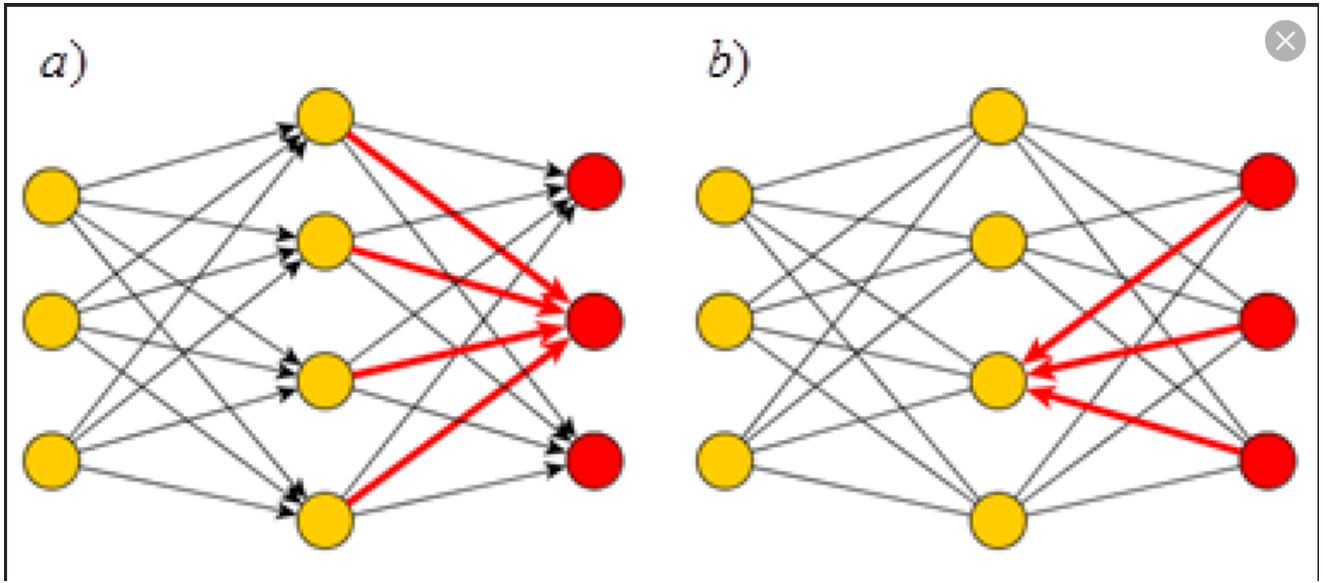


Problems associated with inefficient learning Rate



Forward and Backward Propagation

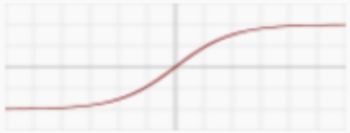
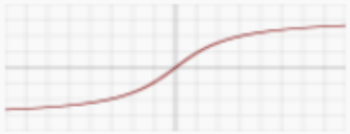
Implementation of forward and backward propagation



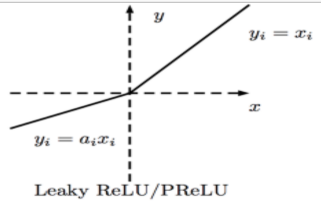
Forward Propagation: mapping from input to output

Backward Propagation: performs a backward pass while adjusting the model's parameters (weights and biases). (For a very good visualization please visit https://www.youtube.com/watch?v=Ilg3gGewQ5U&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi&index=3)

Common Activation Functions

Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$
ArcTan		$f(x) = \tan^{-1}(x)$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$

Leaky ReLu



$$f(x) = \begin{cases} 0.01x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

References

https://www.coursera.org/learn/neural-networks-deep-learning

https://www.deeplearning.ai/ai-notes/optimization/
https://www.deeplearning.ai/ai-notes/initialization/
http://cs231n.stanford.edu/

https://www.youtube.com/c/3blue1brown/playlists

https://medium.com/ravenprotocol/everything-you-need-to-know-about-neural-networks-6fcc7a15cb4
https://towardsdatascience.com/secret-sauce-behind-the-beauty-of-deep-learning-beginners-guide-to-activation-functions-a8e23a57d046