Lecture: Basics of Deep Learning
An Introduction

Dr. Goutam Chakraborty

SAS® Professor of Marketing Analytics

Director of MS in Business Analytics and Data Science* (http://analytics.okstate.edu/mban/)
Director of Graduate Certificate in Business Data Mining (http://analytics.okstate.edu/certificate/grad-data-mining/)
Director of Graduate Certificate in Marketing Analytics (http://analytics.okstate.edu/certificate/grad-marketing-analytics/)

- *Name change pending internal approval.
-
- Some of the slides were developed by Mr. Sanjoy Dey, used with permission.

1

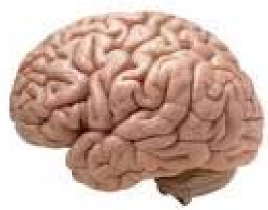
Outline

- Quick recap of traditional neural net used in machine learning (ML)
- What are similarities and differences between machine learning (ML) and deep learning (DL)?

2

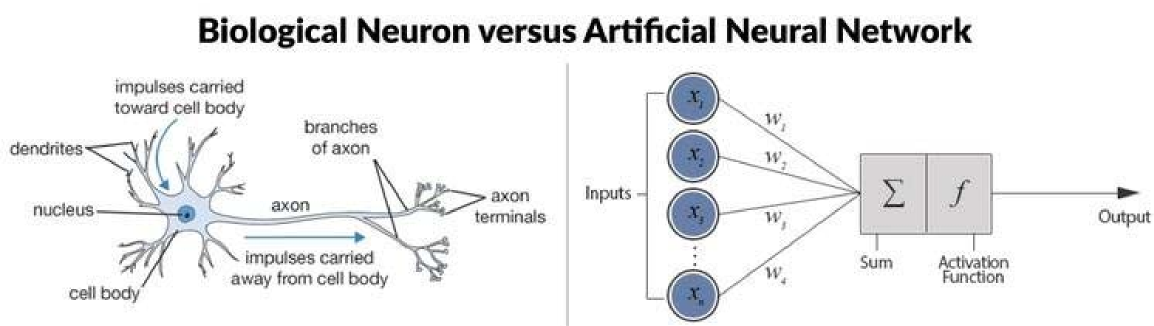
Artificial Neural Net (ANN)

- Developed with the intention to resemble how the human brain works (in particular its ability to learn from experience)!



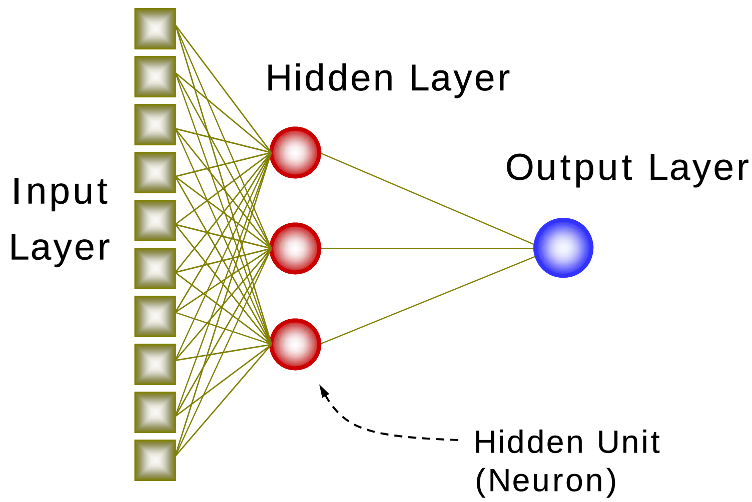
3

Biological Versus Artificial Neural Network



4

Multilayer Perceptron (MLP)



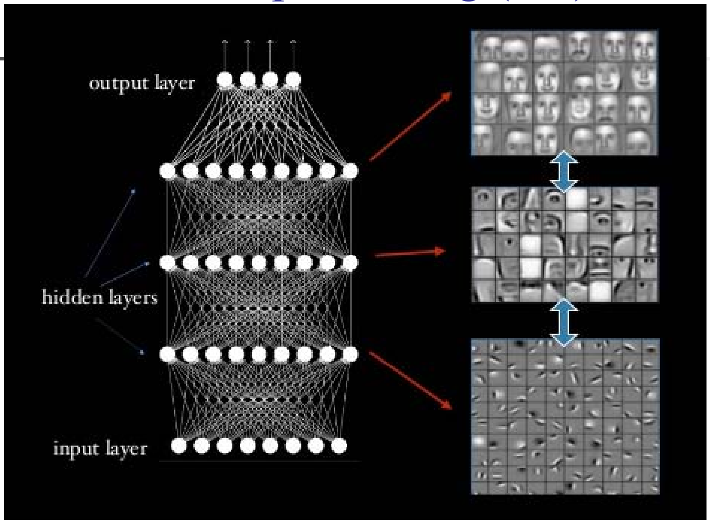
5

Feed Forward Neural Network

- Input data values are passed through from input layer to hidden layer to the output layer
  - Usually all input values are massaged/transformed so that their ranges are restricted to (0,1) or (-1,1)
  - The output value is also restricted to (0,1) but we can always convert it back to its original range
- Values for each weight usually start randomly as each observation is first fed forward through the network
  - Output from feed forward is compared with the actual value and the error is sent backwards
  - The weights are updated (slowly) to see its effect on error
  - Algorithm tries to find optimal weights to minimize overall error

6

Deep Learning (DL)



The term *deep learning* refers to the numerous hidden layers used in a neural network. However, ***the true essence of deep learning is the methods that enable the increased extraction of information*** derived from a neural network with more than one hidden layer.

7

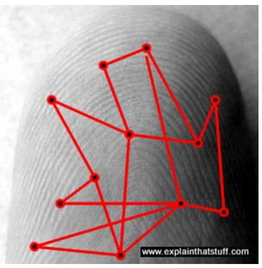
Machine Learning (ML) vs. Deep Learning (DL)

- Feature engineering
- Domain knowledge e.g., finger print reading
- Algorithms
  - SVM
  - Random Forest
  - Logistic regression
  - Decision trees
  - Naïve Bayes
- Deep learning is end-to-end model *without the need for significant domain knowledge and feature engineering*

8

Machine Learning: Feature Engineering

N x N



$f_1, f_2, f_3, \ldots\, f_k$, *where* $k << N$

- In finger print analysis, where the lines terminate or split into two are called minutiae.
- f_1, f_2, could be features such as distance and angle between the lines etc.
- Identifying such features requires domain knowledge

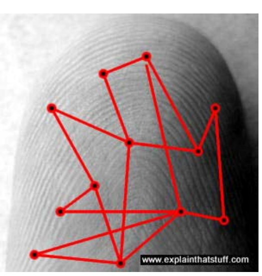
Bob

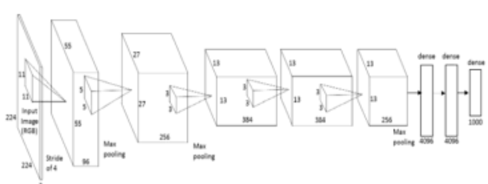
Use your favorite classifier

- SVM
- Random Forest
- Naïve Bayes
- Decision Trees
- Logistic Regression

9

Deep Learning: End-to-End Learning





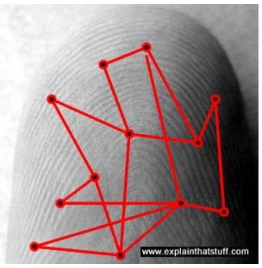
Bob

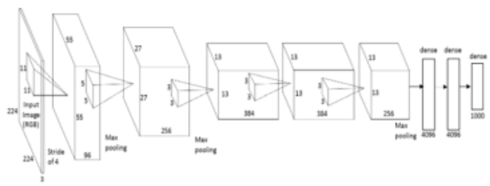
Specific to Deep learning

- Features are learnt by the network
- No need for domain knowledge
- **Lots of data** required to train

10

Deep Learning: End-to-End Learning





Bob

Specific to Deep learning
- Features are learnt by the network
- Little need for domain knowledge
- Lots of data required to train

Common issues in Deep Learning and Machine learning
- Requires **clean** data
- Be careful **not to over fit or, under fit** data during training
- Choose **hyper parameters** carefully
- Choice of **cost function**

11

Lecture: Basics of Deep Learning
Building Blocks

Dr. Goutam Chakraborty

SAS® Professor of Marketing Analytics

Director of MS in Business Analytics and Data Science* (http://analytics.okstate.edu/mban/)
Director of Graduate Certificate in Business Data Mining (http://analytics.okstate.edu/certificate/grad-data-mining/)
Director of Graduate Certificate in Marketing Analytics (http://analytics.okstate.edu/certificate/grad-marketing-analytics/)

- *Name change pending internal approval.
- Note some of these slides are copyrighted by SAS® and used with permission. Reuse or redistribution is prohibited.
- Some of the slides were developed by Mr. Sanjoy Dey, used with permission.

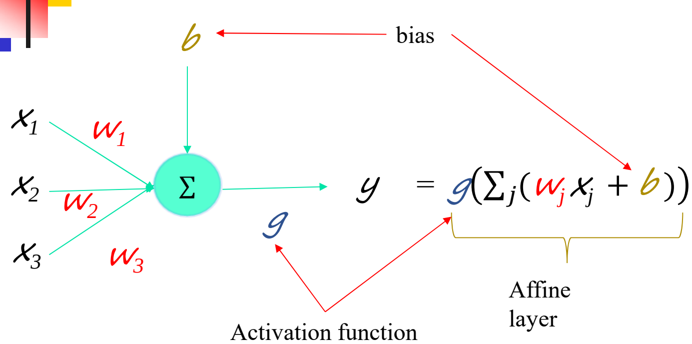
12

Outline

- Traditional Neuron
- Different activation functions
- A bit of math to formalize back propagation
  - Details see Deep learning book by Goodfellow, Bengio and Courville
  - You will see many terms such as: Vectors, Matrices, Tensors and Jacobian…

13

Basic Building Blocks: The Artificial Neuron



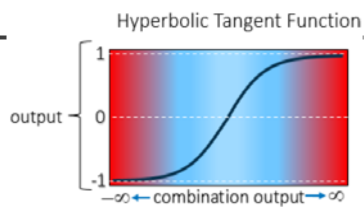
14

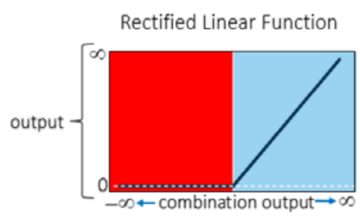
Traditional Neural Networks (ML) Versus Deep Learning (DL)

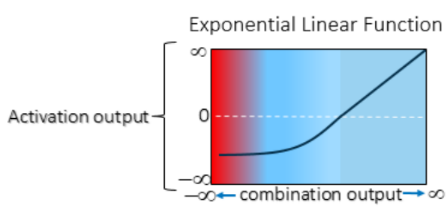
Aspect	Traditional	Deep Learning
Hidden activation function(s)	Hyperbolic Tangent (tanh)	Rectified Linear (RELU) and other variants

15

Activation Functions' Saturation Regions

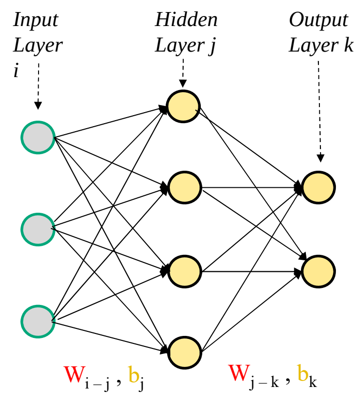






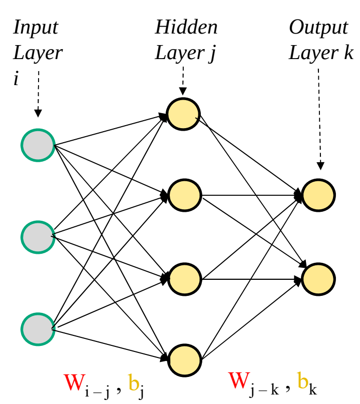
16

A Simple Neural Network



17

A Simple Neural Network for MNIST Dataset



Typical parameters in MNIST dataset:

- Input layer I = 28 X 28 pixels = 784
- Hidden layer j = 100
- Output layer k = 10 (0,1,2..9)
- W_{i-j} = 784 X 100 = 78,400
- b_j = 100
- W_{j-k} = 100 X 10 = 1,000
- b_k = 10

- Total parameters = 79,510

18

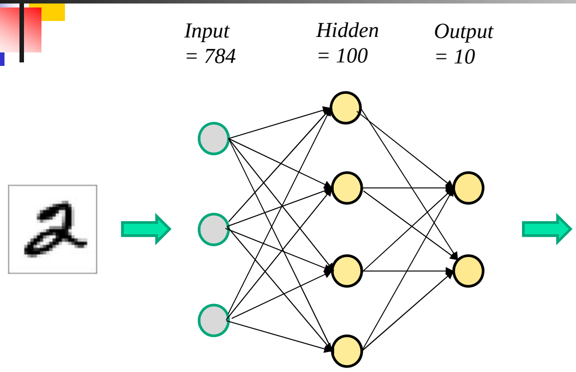
Training a Neural Network

Initialize the parameters → Forward pass through the network and predict an output → Compare output to ground truth and compute a cost → Update the parameters through back propagation → Forward pass through the network and predict an output

Supervised Training

19

Forward Pass



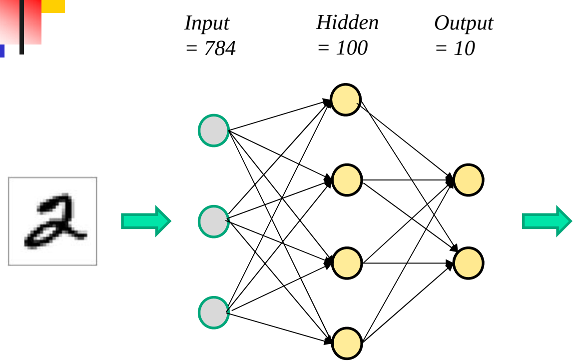
Prediction probability	*Predicted class*
0.0	0.
0.3	1
0.1	2
0.0	3
0.0	4
0.0	5
0.1	6
0.4	**7**
0.0	8
0.1	9

Prediction probability = Softmax Activation (Output)

$$\text{Softmax function} = \sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

20

Cost Functions



Prediction probability	Ground truth
0.0	0.0
0.1	0.0
0.3	**1.0**
0.0	0.0
0.0	0.0
0.0	0.0
0.1	0.0
0.4	0.0
0.0	0.0
0.1	0.0

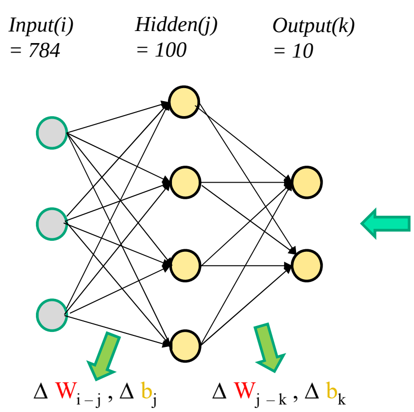
Cost function = C(output prediction, ground truth)

Typical cost functions are:

- Mean Squared Error
- Cross Entropy Loss

21

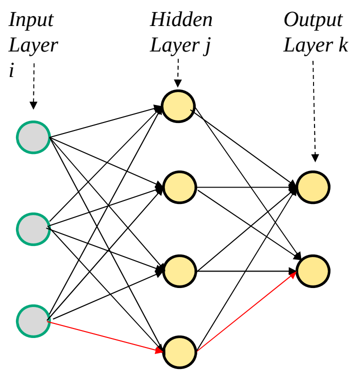
Back Propagation



Prediction probability	Ground truth
0.0	0.0
0.1	0.0
0.3	**1.0**
0.0	0.0
0.0	0.0
0.0	0.0
0.1	0.0
0.4	0.0
0.0	0.0
0.1	0.0

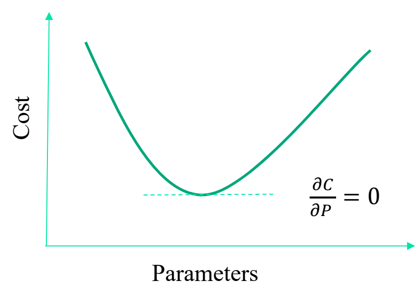
22

Back Propagation



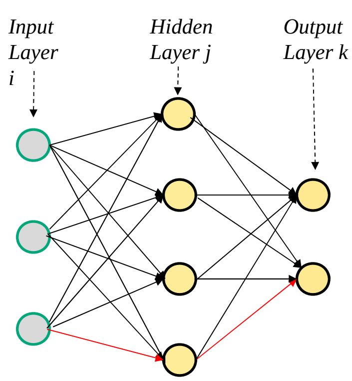
Compute $\frac{\partial(Cost=C)}{\partial(parameters=P)}$

Gradient Descent Algorithm



23

Back Propagation (Contd.)



Compute $\frac{\partial(Cost=C)}{\partial(parameters=P)}$

$$C(y, truth) = C(g_k(\Sigma_k(w_k x_k + b_k)))$$

$$a_k = \Sigma_k(w_k x_k + b)$$

$$C(y, truth) = C(g_k(a_k(w_k, x_k)))$$

$$\frac{\partial C}{\partial P} = \frac{\partial C}{\partial g_k} \cdot \frac{\partial g_k}{\partial a_k} \cdot \frac{\partial a_k}{\partial P}$$

$$x_k = g_j(\Sigma_j(w_j x_j + b_j))$$

$$x_k = g_j(a_j(w_j, x_j))$$

$$C(y, truth) = C(g_k(a_k(w_k, g_j(a_j(w_j, x_j)))))$$

$$\frac{\partial C}{\partial P} = \frac{\partial C}{\partial g_k} \cdot \frac{\partial g_k}{\partial a_k} \cdot \frac{\partial a_k}{\partial g_j} \cdot \frac{\partial g_j}{\partial a_j} \cdot \frac{\partial a_j}{\partial P}$$

24

Lecture: Basics of Deep Learning
Training Issues

Dr. Goutam Chakraborty

SAS® Professor of Marketing Analytics

Director of MS in Business Analytics and Data Science* (http://analytics.okstate.edu/mban/)
Director of Graduate Certificate in Business Data Mining (http://analytics.okstate.edu/certificate/grad-data-mining/)
Director of Graduate Certificate in Marketing Analytics (http://analytics.okstate.edu/certificate/grad-marketing-analytics/)

- *Name change pending internal approval.
-
- Some of the slides were developed by Mr. Sanjoy Dey, used with permission.

25

Outline

- Weight initializations
- Regularizations

26

Traditional Neural Networks (ML) Versus Deep Learning (DL)

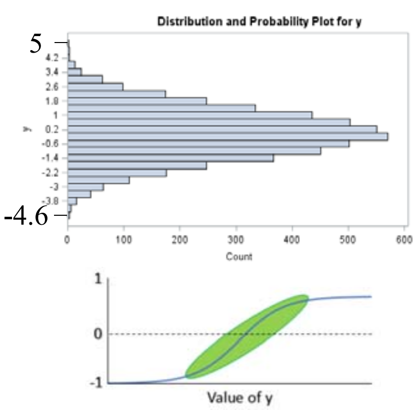
Aspect	Traditional	Deep Learning
Hidden activation function(s)	Hyperbolic Tangent (tanh)	Rectified Linear (RELU) and other variants
Weight initialization	Constant Variance	Normalized Variance

27

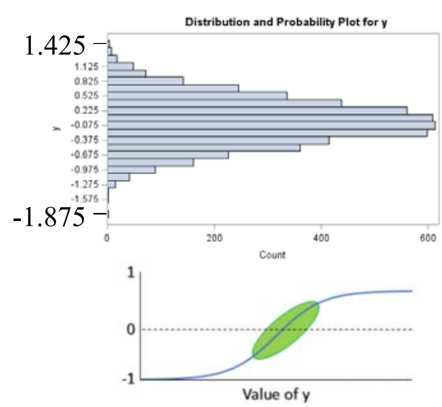
Weight Initializations

Consider: $y = w_1 + \ldots + w_{25}$

- Constant Variance
- (Standard deviation=1)



Normalized Variance
(Standard deviation=$\sqrt{\frac{6}{25+25}} \approx .34$)



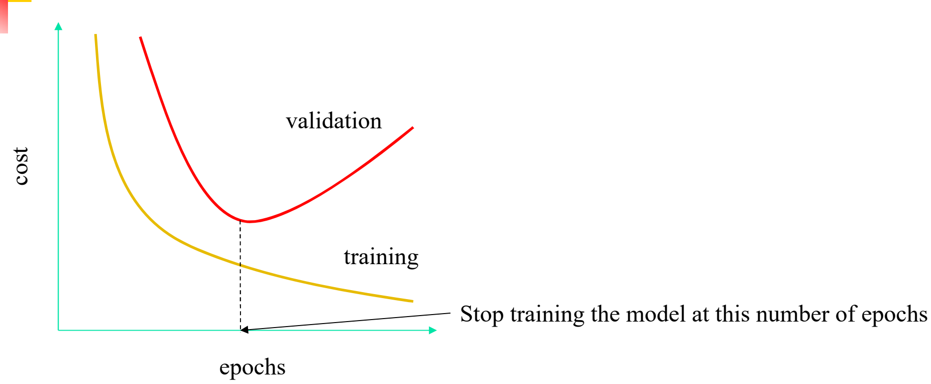
28

Traditional Neural Networks (ML) Versus Deep Learning (DL)

Aspect	Traditional	Deep Learning
Hidden activation function(s)	Hyperbolic Tangent (tanh)	Rectified Linear (RELU) and other variants
Weight initialization	Constant Variance	Normalized Variance
Regularization	Stopped Training, L1, and L2	Stopped Training, L1, L2, Dropout, and Batch Normalization

29

Managing Over Fitting: Early Stopping

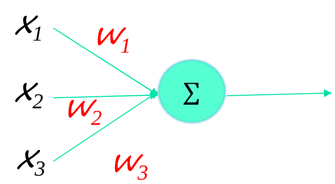


30

Managing Over Fitting: L2 Regularization

L2 regularilzation

$$\text{Cost} = \underbrace{\tfrac{1}{2}\,\Sigma(y_{pred} - y_{truth})^2}_{\text{Mean Square Error}} + \underbrace{\lambda}_{\text{moderator}} * \underbrace{W^T W}_{\text{L2 regularizor}}$$

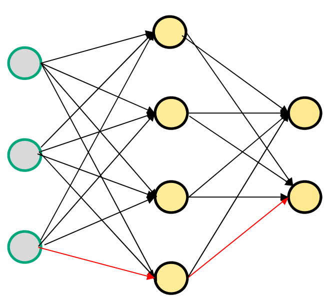


Non-regularized situation → $W_1 >> 1 \quad W_2 \sim 0 \quad W_3 \sim 0$

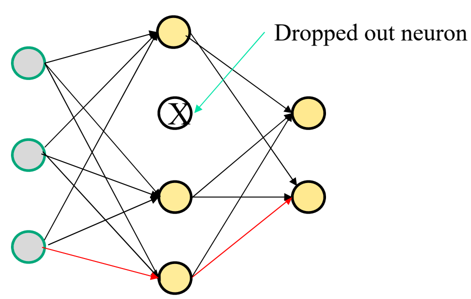
Regularized situation → W_1 , W_2 , W_3 *are in similar range of values*

31

Managing Over Fitting: Dropout Regularization



Normal network



Network with dropout

Dropout is applied only during training. In testing all parameters are used. Benefits of dropout are:

- All parameters are forced to have an impact
- Computationally inexpensive

32

Lecture: Basics of Deep Learning
Training Efficiency

Dr. Goutam Chakraborty

SAS® Professor of Marketing Analytics

Director of MS in Business Analytics and Data Science* (http://analytics.okstate.edu/mban/)
Director of Graduate Certificate in Business Data Mining (http://analytics.okstate.edu/certificate/grad-data-mining/)
Director of Graduate Certificate in Marketing Analytics (http://analytics.okstate.edu/certificate/grad-marketing-analytics/)

- *Name change pending internal approval.
-
- Some of the slides were developed by Mr. Sanjoy Dey, used with permission.

33

Outline

- Gradient Descent and its variants
- Mini batches
- Batch normalization
- CPU vs. GPU
- Hyperparameters: Data scientist’s expertise

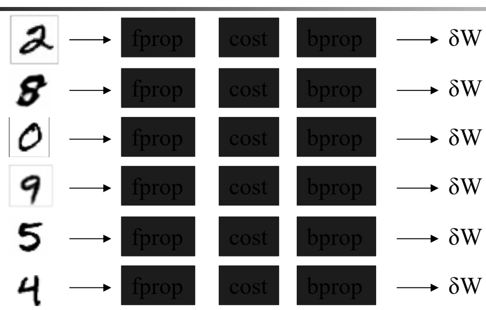
34

Traditional Neural Networks (ML) Versus Deep Learning (DL)

Aspect	Traditional	Deep Learning
Hidden activation function(s)	Hyperbolic Tangent (tanh)	Rectified Linear (RELU) and other variants
Weight initialization	Constant Variance	Normalized Variance
Regularization	Stopped Training, L1, and L2	Stopped Training, L1, L2, Dropout, and Batch Normalization
Gradient-based learning	Batch GD and BFGS	Stochastic GD, Adam, and LBFGS

35

Enhancing Training Efficiency: Mini Batch



Inefficient process :
Parameters updated after all inputs are processed.

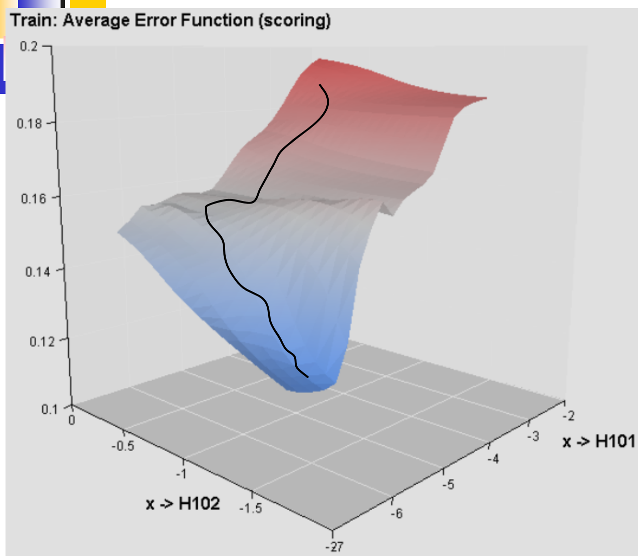
$$\Delta W = \alpha \frac{1}{N} \Sigma \delta W$$

learning rate

This is **normal Gradient Descent**

36

Batch Gradient Descent

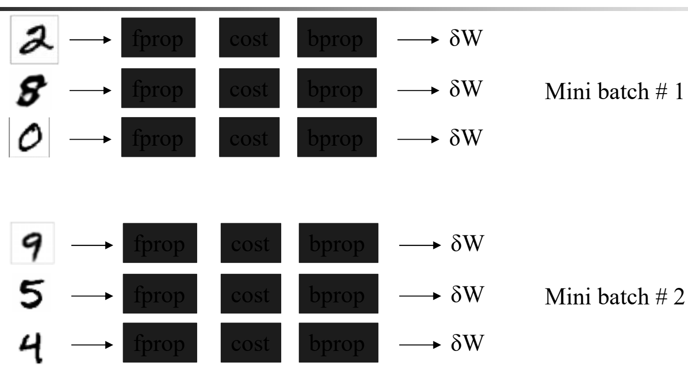


$$\delta^{(t)} = -\eta \nabla g^{(t)} + \alpha \delta^{(t-1)}$$

- Uses **all of the training observations (t)** to calculate the exact gradient on each descent step
- Results in a smooth progression to the error minima

37

Enhancing Training Efficiency: Mini Batch (Contd.)

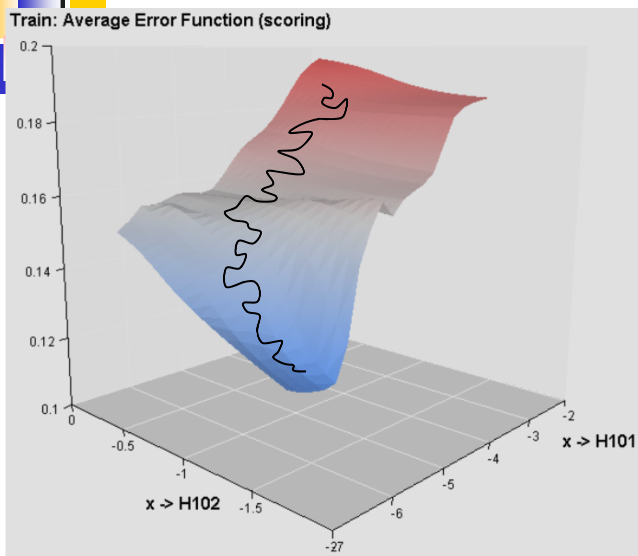


Efficient process :
Parameters updated after each mini batch is processed

This is called **Stochastic Gradient Descent**

38

Stochastic Gradient Descent



$$\boxed{\delta^{(i)} = -\eta \nabla g^{(i)} + \alpha \delta^{(i-1)}}$$

- Uses a single training (i) observation to calculate an approximate gradient for each descent step
- Results in a chaotic progression to the error minima **but faster than GD**

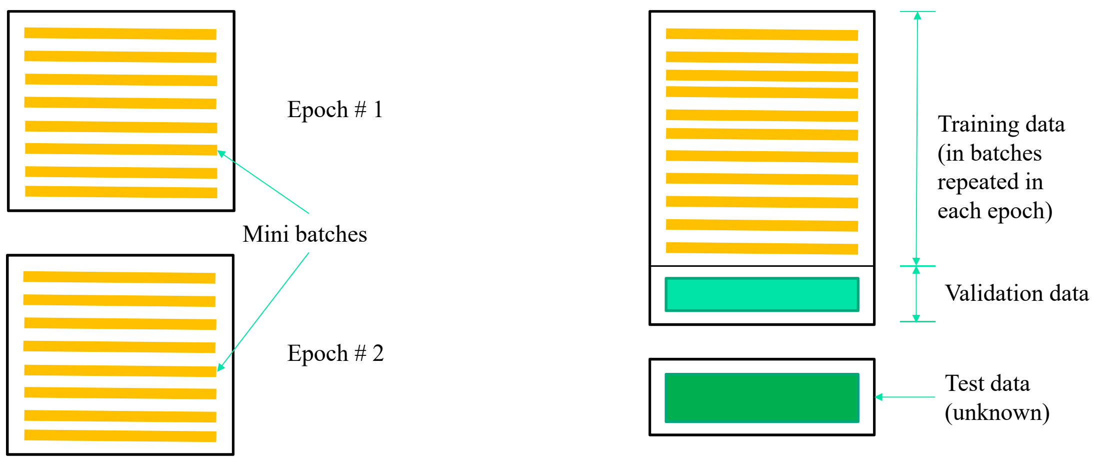
39

ADAM Optimization

- The ADAM method applies adjustments to the learned gradients for each individual model parameter in an *adaptive manner* by approximating second-order information about the objective function based on previously observed mini-batch gradients.
- The ADAM method introduces two new **hyperparameters** to the mix, (β^t_1) and (β^t_2) where t represents the iteration count.
  - The adjustable beta terms are used to approximate a *signal-to-noise* ratio that is used to scale the step size.
  - When the approximated single-to-noise ratio is small, the step size is near zero.
- A learning rate, α , is also included in the optimization method

40

Enhancing Training Efficiency: Epochs



41

Batch Normalization

- Standardizes each piece of input data by subtracting its mean and dividing by its standard deviation
- It then follows this calculation by *multiplying* the data by the value of a *learned constant* and then *adding* the value of *another learned constant*

$$\gamma * \left(\frac{X_i - \mu}{\sigma}\right) + \beta$$

42

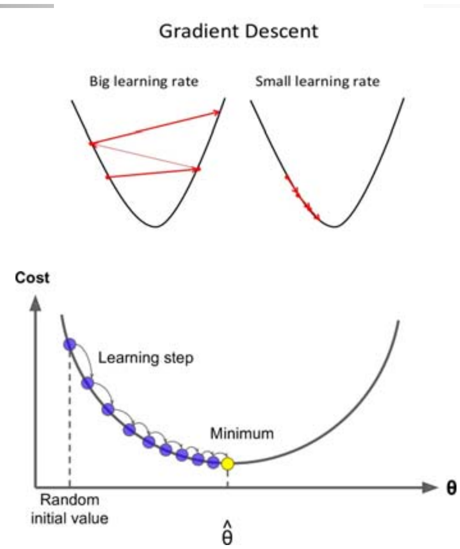
Traditional Neural Networks (ML) Versus Deep Learning (DL)

Aspect	Traditional	Deep Learning
Hidden activation function(s)	Hyperbolic Tangent (tanh)	Rectified Linear (RELU) and other variants
Weight initialization	Constant Variance	Normalized Variance
Regularization	Stopped Training, L1, and L2	Stopped Training, L1, L2, Dropout, and Batch Normalization
Gradient-based learning	Batch GD and BFGS	Stochastic GD, Adam, and LBFGS
Processor	CPU	GPU

43

Increasing Accuracy : Choosing Hyperparameters

- Model structure (expertise needed)
  - Number of hidden layers
  - Size of each hidden layer
  - Initialization of weights
  - Choice of optimizer
  - Activation functions
  - Loss function
- Scalar parameters
  - Number of epochs
  - Learning rate α
  - Regularization moderator
    - Dropout probability p
    - L2 moderator λ
  - Batch size



44