Supervised Learning: Overview and Deep Learning Advancements

Earl Wong

Learning

Supervised Learning

Unsupervised Learning

Reinforcement Learning

Learning

Supervised Learning

Learning With Labeled Training Data: [Input Data, Label]

IRIS Input Data

Sepal length [cm] = 5.1
Sepal width [cm] = 3.5
Petal length[cm] = 1.4
Petal width [cm] = 0.2

Pascal VOC Input Data

Label

Setosa

Label

Car

Learning

Supervised Learning

Underlying Objective

IRIS Input Data

Sepal length [cm] = 4.9
Sepal width [cm] = 3.0
Petal length[cm] = 1.5
Petal width [cm] = 0.2

→ AlgorithmA →

Label

Setosa

Pascal VOC Input Data

→ AlgorithmB →

Label

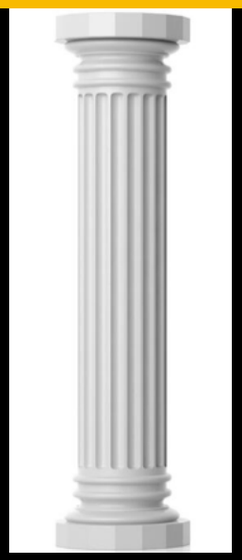
Bicycle

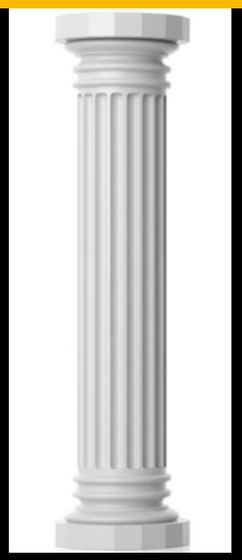
Learning Example
NN Based Classifier

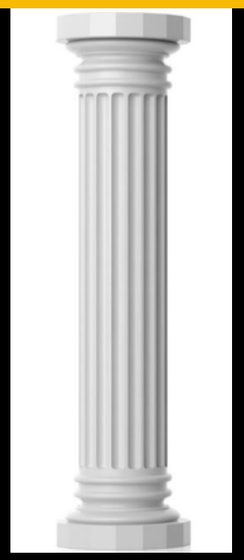
Labeled Dataset

Algorithm

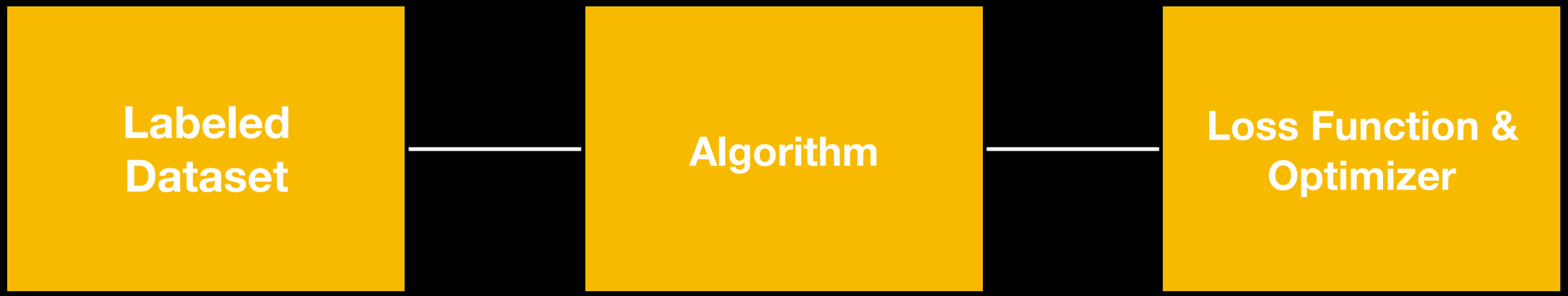
Loss Function & Optimizer







Learning Example
NN Based Classifier



Knowledge from the labeled dataset is imparted into an algorithm using a loss function and backpropagation.

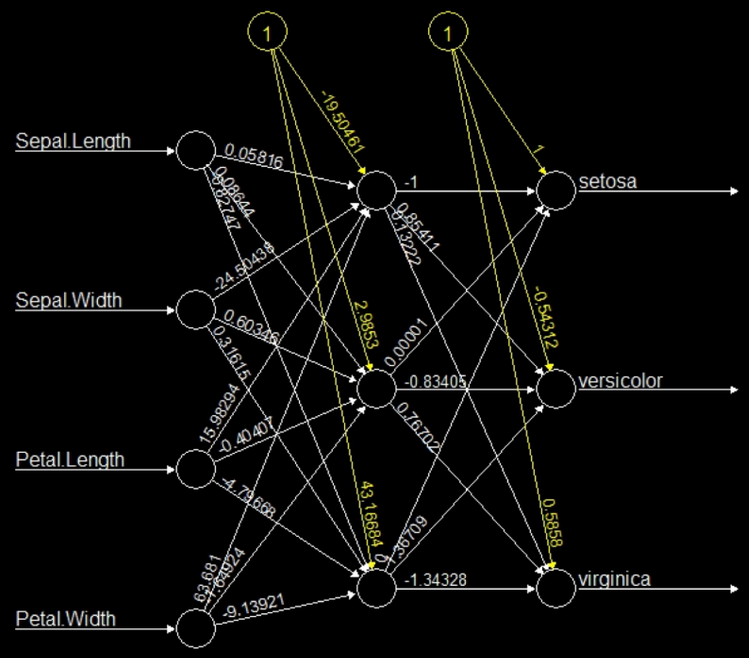
The loss function helps “teach” the algorithm what it needs to learn / know.

Learning Example
NN Based Classifier (Classical)

Dataset: IRIS
Input: 1x4
Labels: 3
Examples: 150

Algorithm
(Architecture)
MLP
Weights: 21

Loss Function:
MSE
Optimizer: SGD

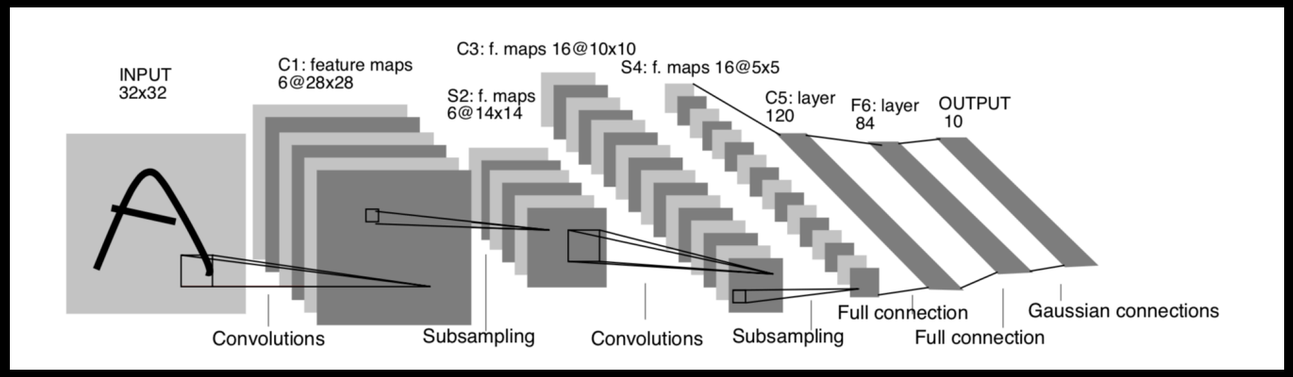


Learning Example
NN Based Classifier (Classical Vision)

Dataset: MNIST
Input: 32x32
Labels: 10
Examples: 70,000

Algorithm
(Architecture)
CNN & FC
Weights: 60,000
(LeNet5)

Loss Function:
MSE
Optimizer: SGD

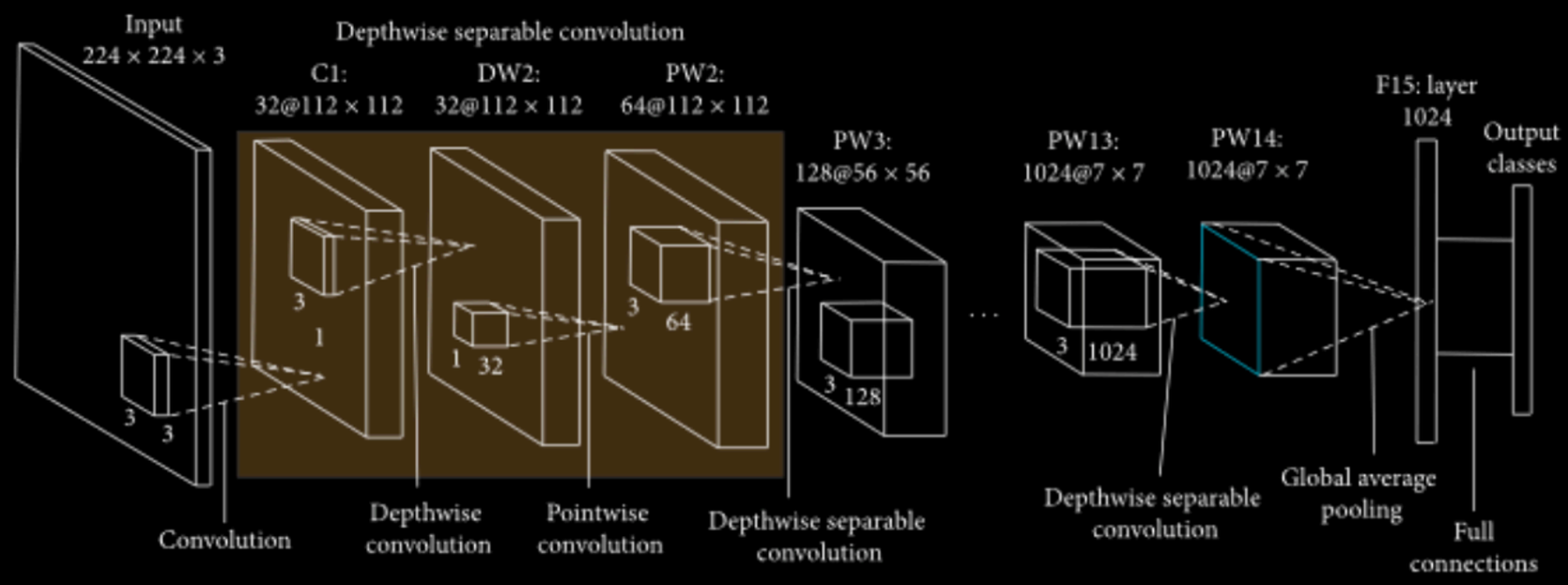


Learning Example
ML Classifier
(Modern Day Vision)

Dataset: Imagenet
Input: 3x224x224
Labels: 22,000
Examples: 14 million

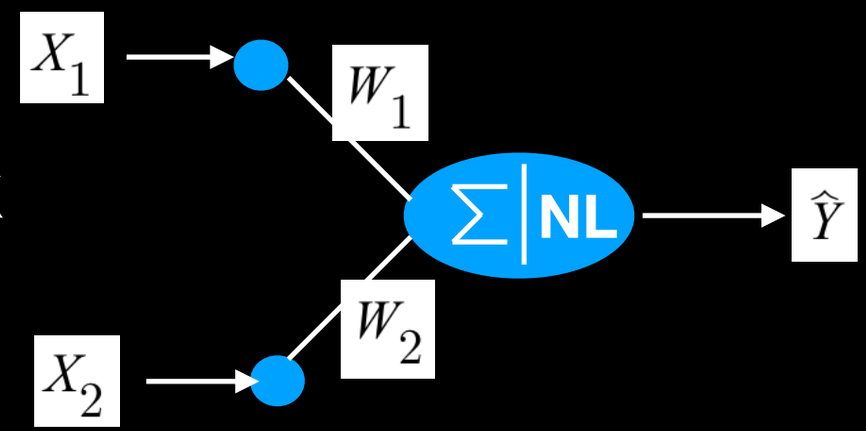
Algorithm (Architecture)
CNN & FC
Weights: >= 500,000
(.25 Mobilenet-224)

Loss Function: Cross Entropy
Optimizer: ADAM (SGD + momentum + adaptive learning rate + ….)



Fundamental Building Blocks

- **Artificial Neural Network**



- **Loss Function**

$$L = \left(y - f(x)\right)^2$$

- **Back propagation**

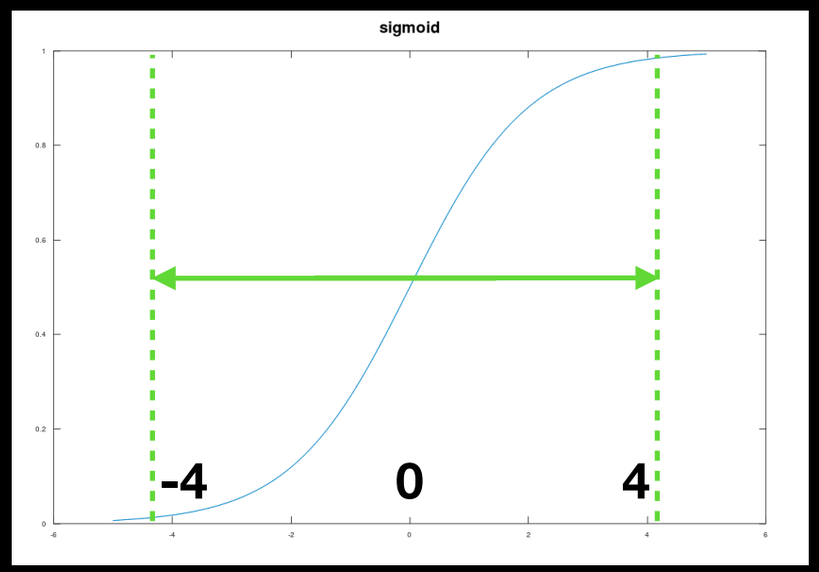
$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Important Advances

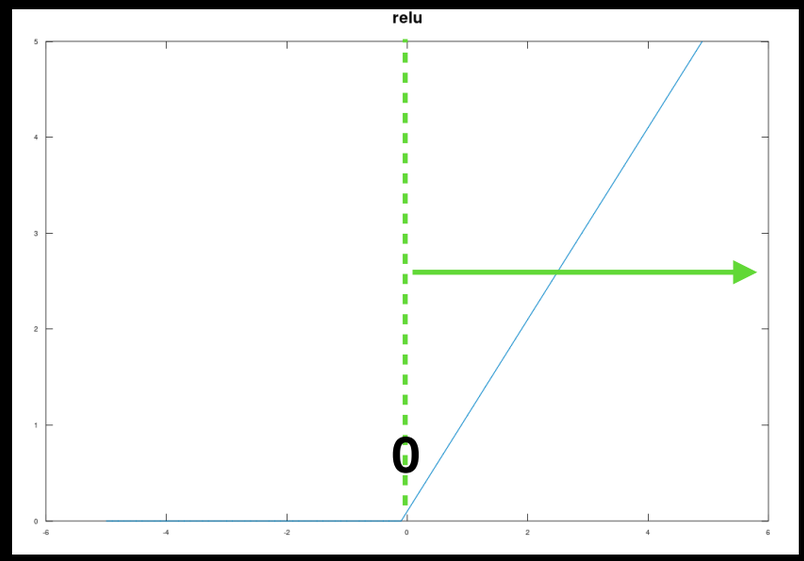
- NL Activation Unit
- Loss Function
- Optimizer
- Batch Normalization
- Dropout

ANN Evolution - NL Activations Units

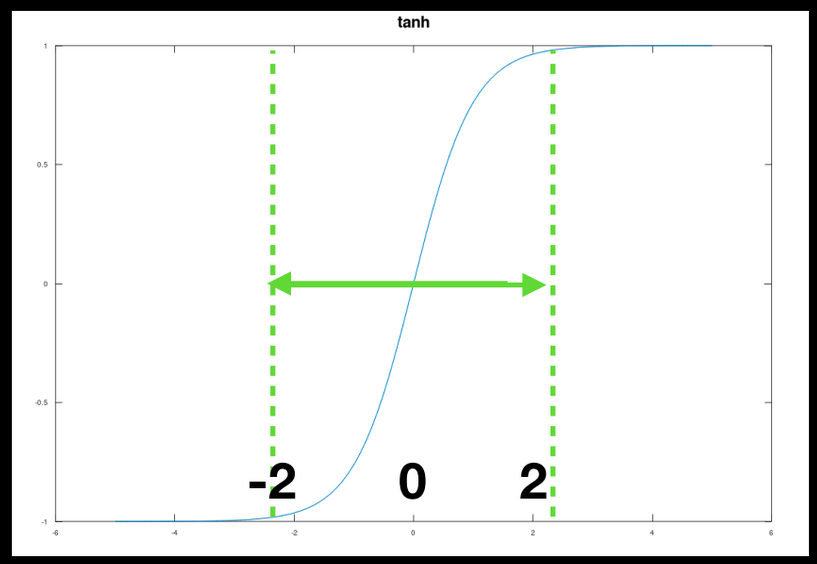
Classical: Sigmoid



Current: Relu



Classical: TanH



The green arrow denote the non-saturating range for different activation functions.

The vanishing gradient problem is less severe for Relu's because saturation cannot occur for positive inputs.

This enabled the training of deeper networks.

However, since no output is produced for negative inputs many dead neurons exist in neural nets employing Relu activations.

Loss Function Evolution

Classical	Current
MSE Loss (L2 loss)	Cross Entropy Loss (CE)
MAE Loss (L1 loss)	Triplet Loss
	Multitask Loss (CE + MSE or MAE)
	Discriminator of a GAN

Backpropagation Evolution

Classical

Gradient Descent

Current

Add Momentum (i.e. Create a running average of previously computed gradients)

Add Adaptivity (i.e. Adaptively adjust the learning rate by dividing the learning rate by the magnitude of the running average)

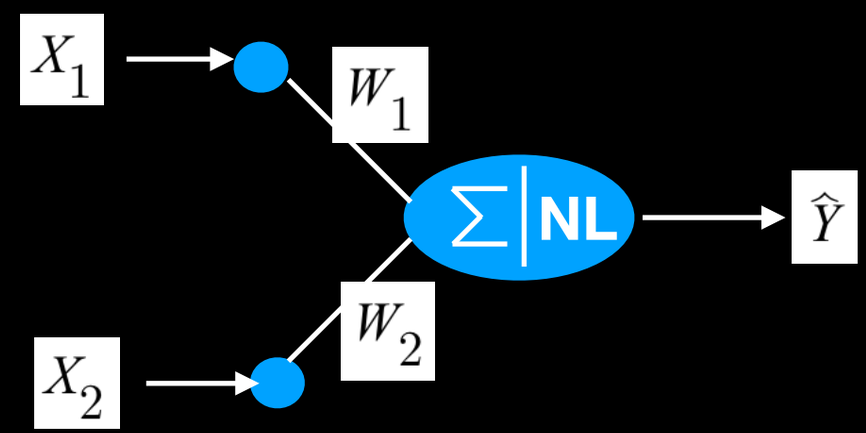
ADAM (Adaptive Momentum)

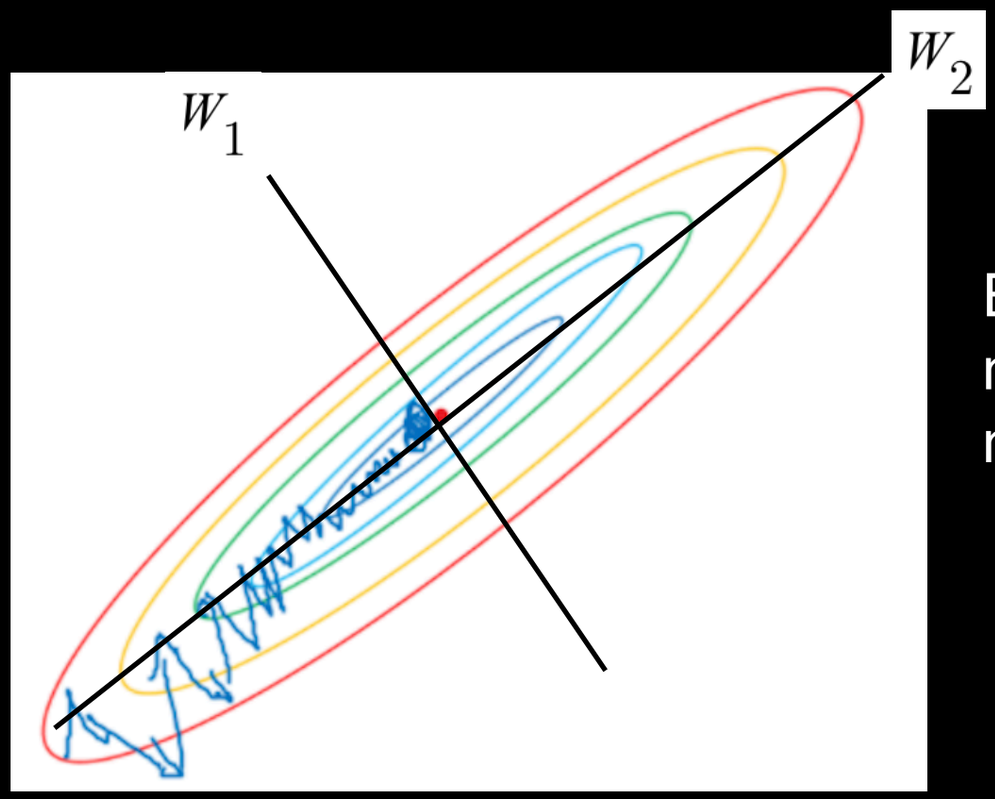
Old Ideas / New Beginnings

- The power of normalization [BatchNormalization]
- The power of ensembles [Dropout]

The Power of Normalization

X_1 = height of males [km]
X_2 = height of females [mm]

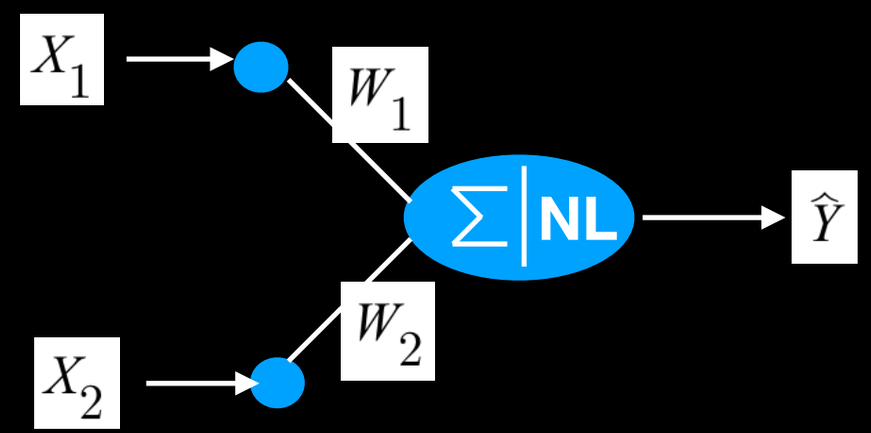


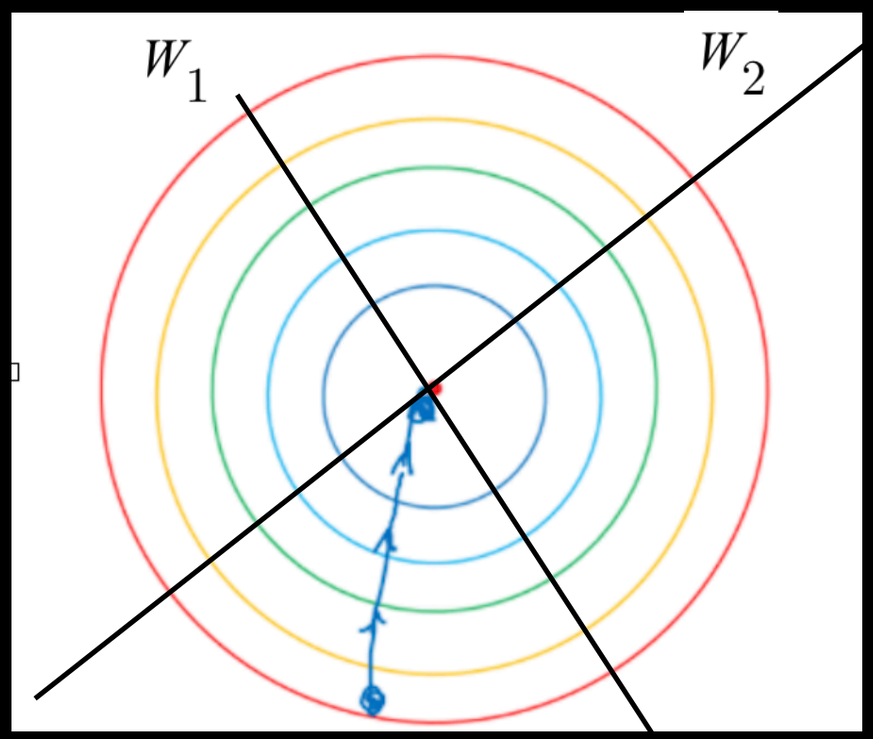


By choosing inappropriate input units, you can make the optimization problem significantly more challenging.

The Power of Normalization

X_1 = height of males [m]
X_2 = height of females [m]





By normalizing the inputs, the becomes significantly easier.

Batch Normalization

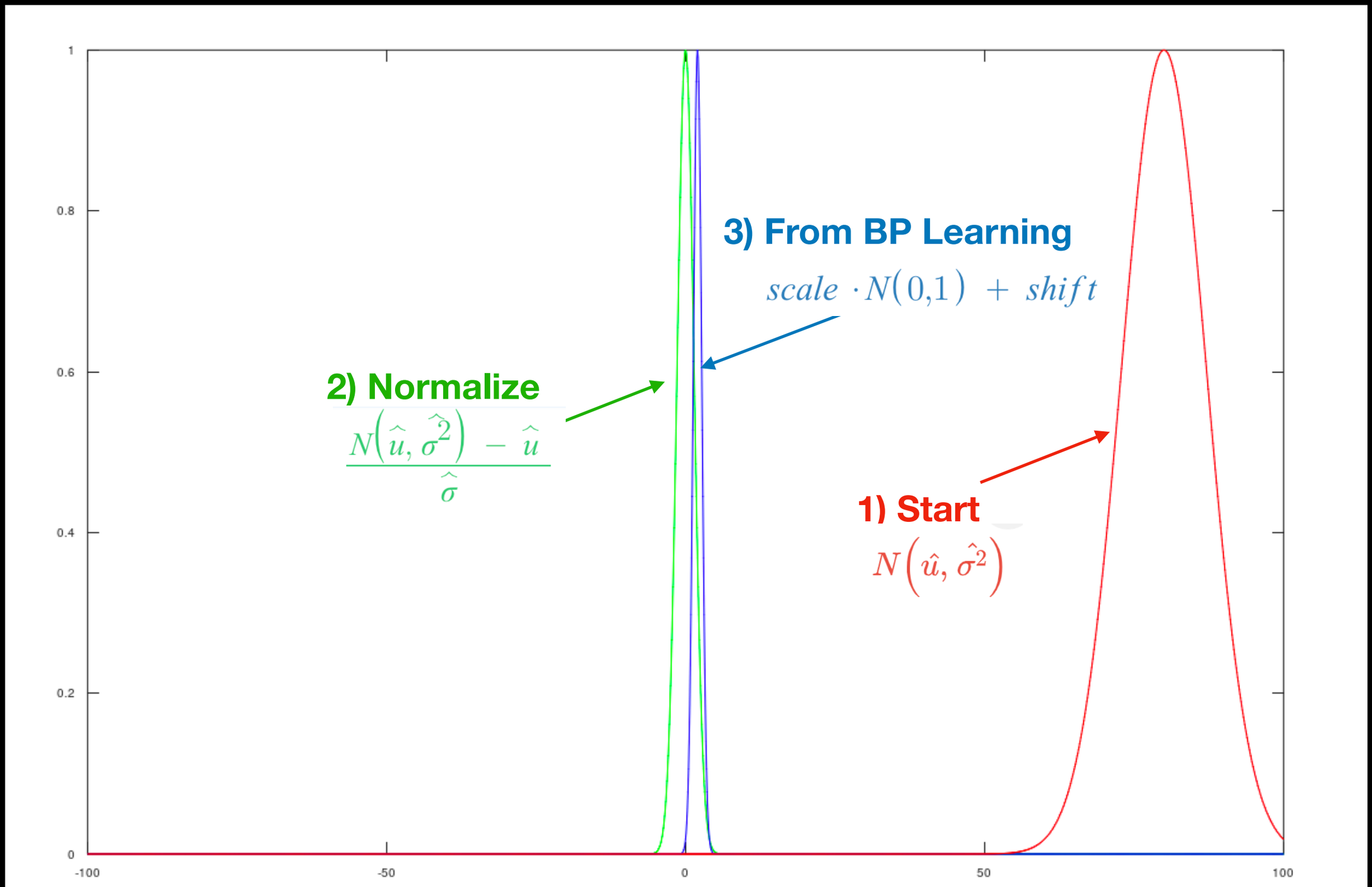
Batch normalization normalizes each neuron output with its empirical mean $\hat{u}$ and empirical standard deviation $\hat{\sigma}^2$ computed form the input mini batch: $N\left(\hat{u}, \hat{\sigma^2}\right)$

The neuron outputs are then rescaled to $N(0,1)$ distributions: $\frac{N\left(\hat{u}, \hat{\sigma^2}\right) - \hat{u}}{\hat{\sigma}}$

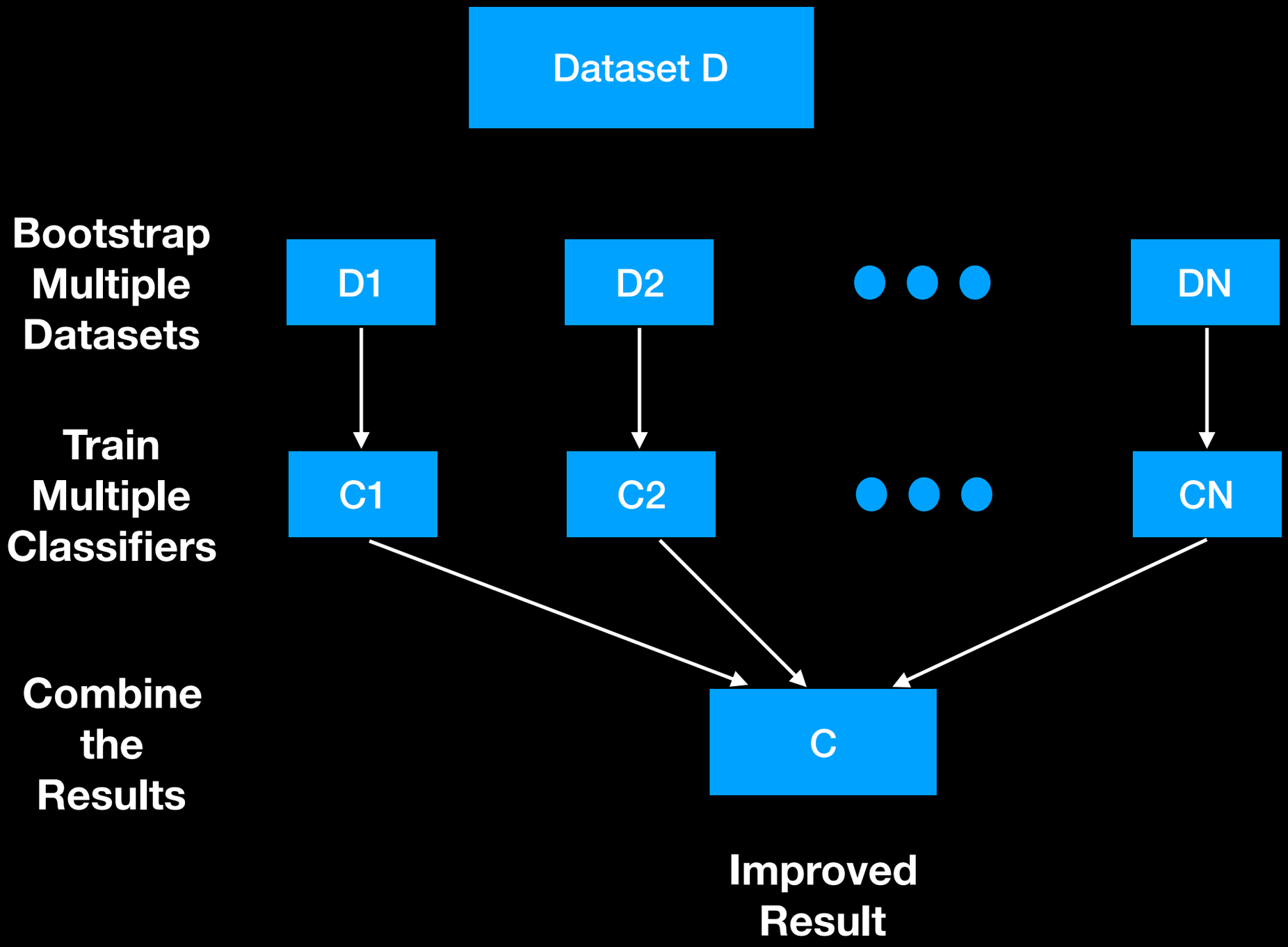
In addition, batch normalization allows the network to learn parameters that subsequently scale and shift the normalized distributions: $scale \cdot N(0,1) + shift$

Benefits: This results in faster learning and the ability to train deeper networks.

Batch Normalization Objective



The Power of Ensembles



The Power of Ensembles

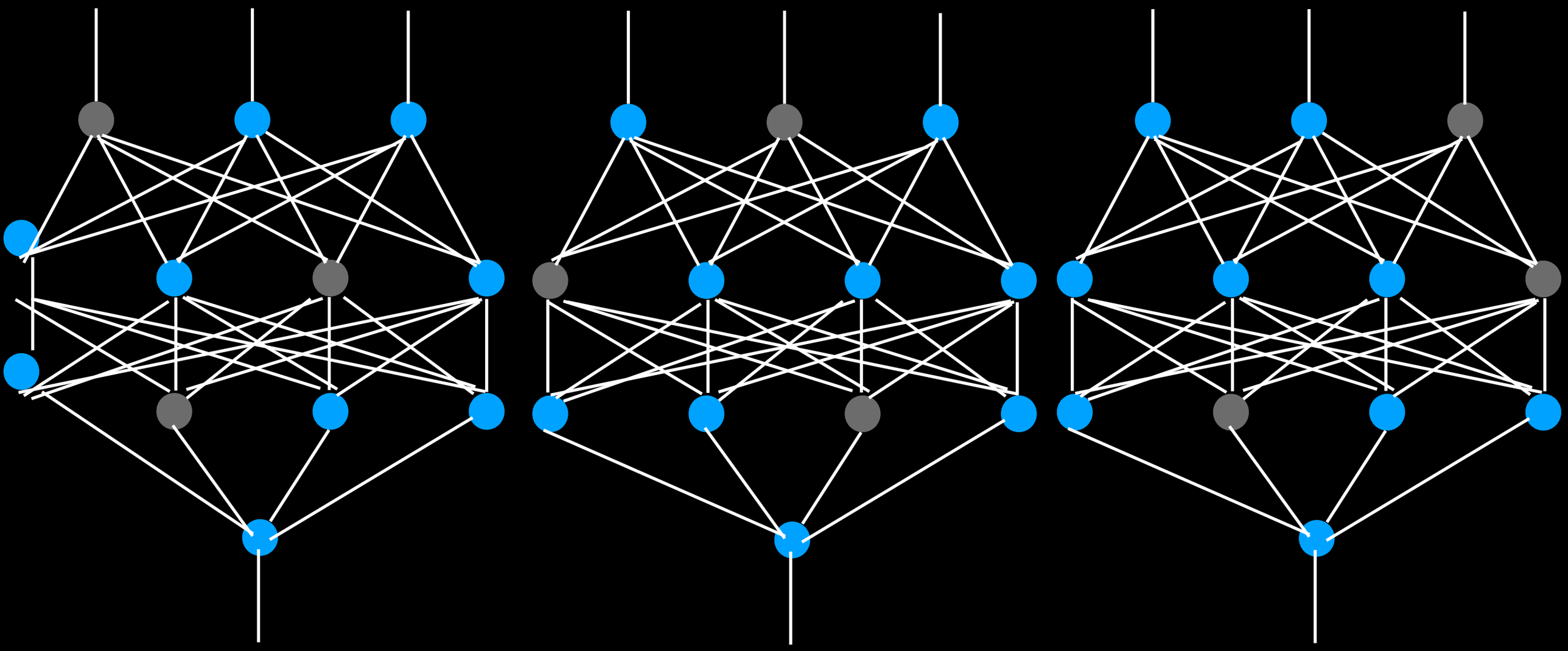
- Dropout seeks to achieve similar advantages.
- During training, neurons are removed from the network.
- This has the effect of creating “pseudo” ensembles networks, WITHIN a single network.
- As a result, overfitting is reduced and generalization is improved.

Dropout

During miniBatch 1, 4, …
dropout these gray nodes
during training

During miniBatch 2, 5, …
dropout the gray nodes
during training

During miniBatch 3, 6, …
dropout these gray nodes
during training



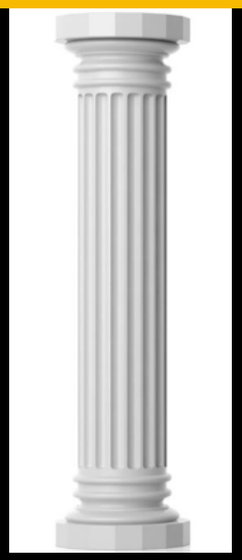
Nodes / neurons where signal cannot flow.

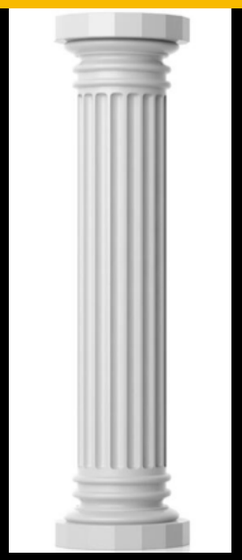
Next: A Deeper Dive

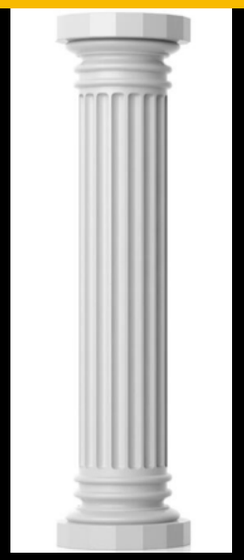
Labeled Dataset

Algorithm

Loss Function







References

- The Perceptron, Rosenblatt, Psychological Review 1958
- Learning Representations by Backpropagation Errors, Rumelhart, et. al., Nature 1986
- Adam: A Method for Stochastic Optimization, Kingma, et. al., arXiv December 2014
- Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, arXiv February 2015
- Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Srivastava, et. al. JMLR 2014