What is Deep Learning

Ben Heil
10/29/2020

Housekeeping

Questions go in the chat or on google doc

Tweet with the #DLBio hashtag

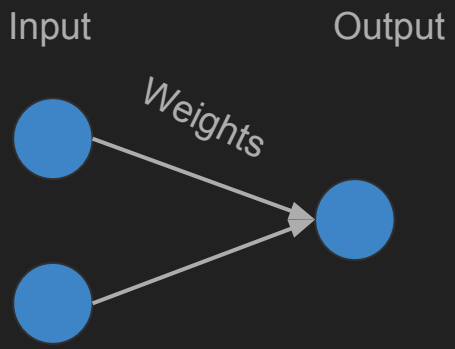
Workshop is being recorded

I am...

- A 3rd year PhD candidate at UPenn in the Casey Greene lab
- Currently researching multitask training on large transcriptomic datasets
- A blogger who posts about ML/Reproducible Research at ben-heil.github.io/
- A pseudobiologist

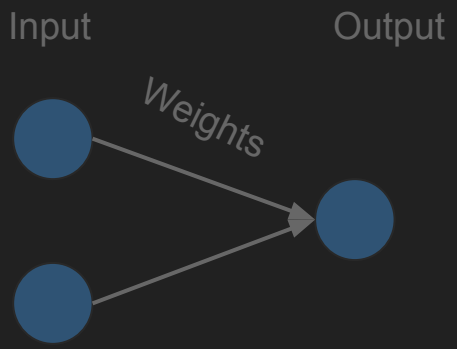
What is Deep Learning

Linear Regression

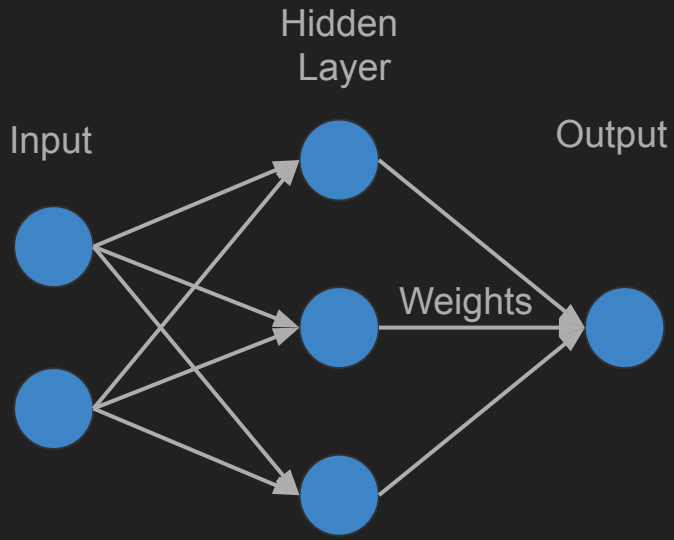


What is Deep Learning

Linear Regression



Deep Learning



Deep Learning Methods are Characterized By

Learning their own features

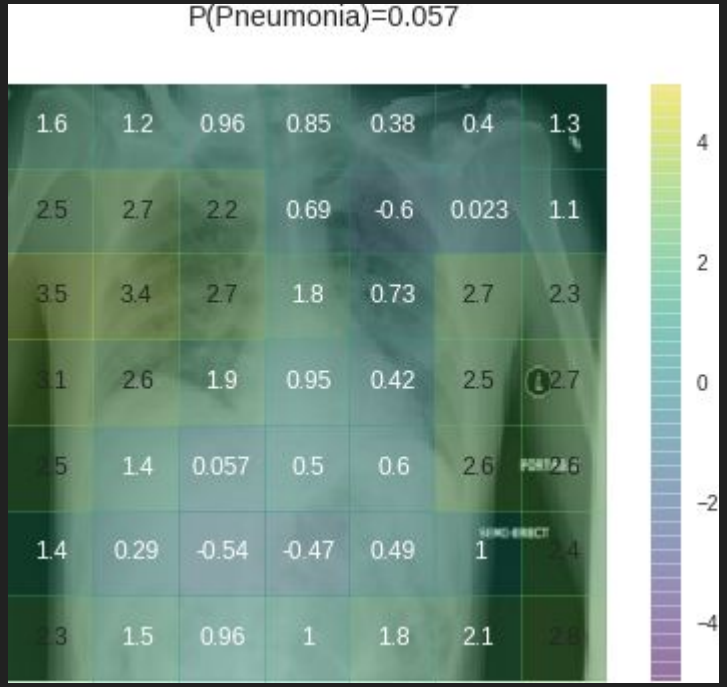
Neural network models

Huge capacity for improving with more compute/data

Why to Not Use Deep Learning

1. You must have a ton of data
2. The model may learn things from the data that you don’t want
3. Logistic/linear regression can do better in a lot of cases

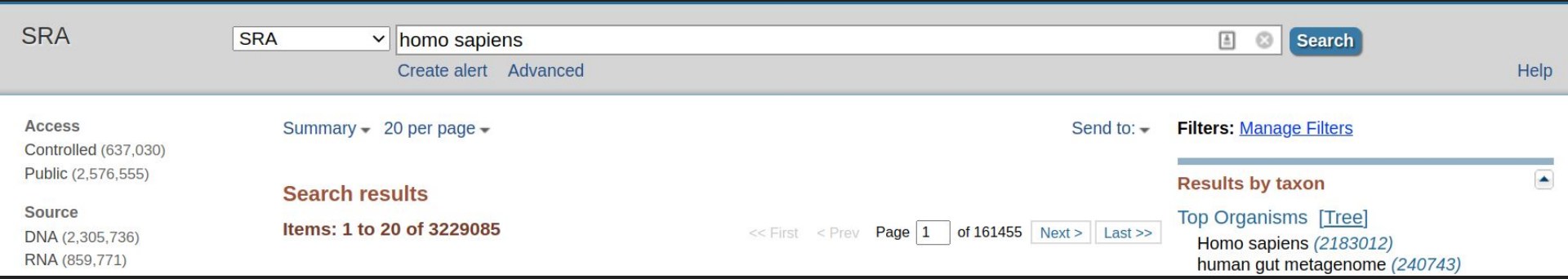
Why to Not Use Deep Learning



https://medium.com/@jrzech/what-are-radiological-deep-learning-models-actually-learning-f97a546c5b98

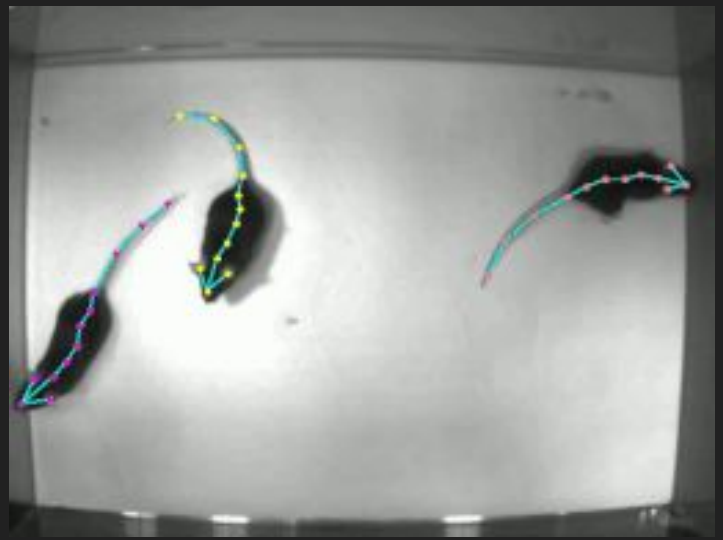
Why to Use Deep Learning

We have data



Why to Use Deep Learning

Using DL can save scientist-hours and make previously intractable questions easy



http://www.mousemotorlab.org/deeplabcut

Why to Use Deep Learning

Cool method names

ELMo (https://arxiv.org/abs/1802.05365)

BERT (https://arxiv.org/abs/1810.04805)

Big Bird (https://arxiv.org/abs/2007.14062)

Neocognitron (https://link.springer.com/article/10.1007/BF00344251)

(Some) Areas of Active Deep Learning Research

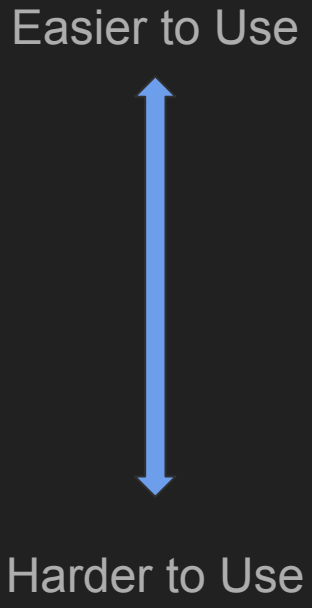
Image processing

Natural language processing (NLP)

Adversarial Learning

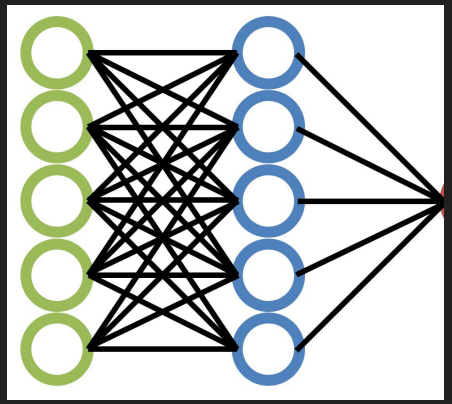
Simulating data

Robust ML

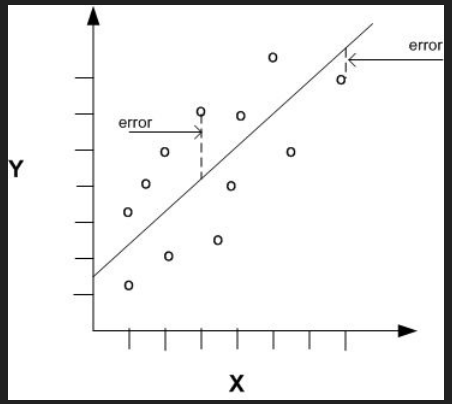


Roadmap

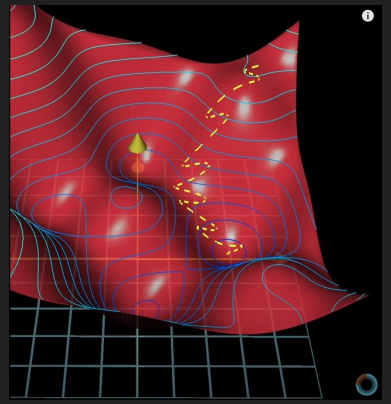
Model



Loss



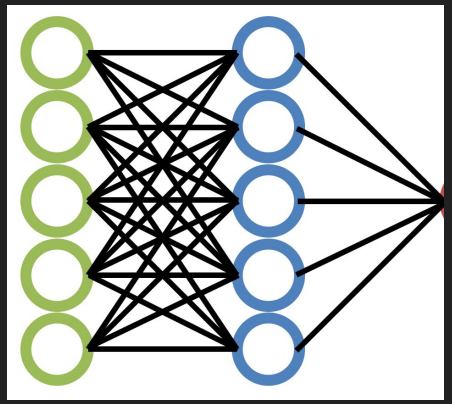
Optimizer



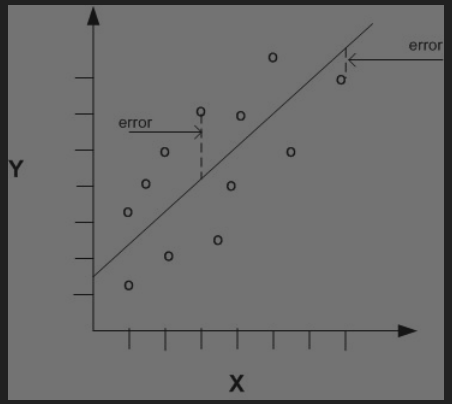
Data

Roadmap

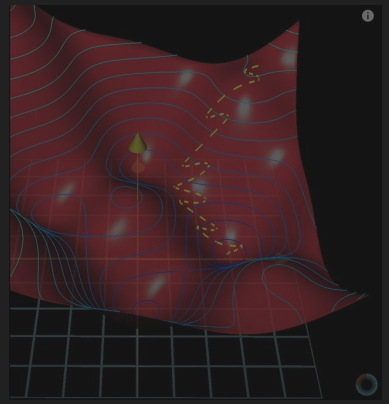
Model



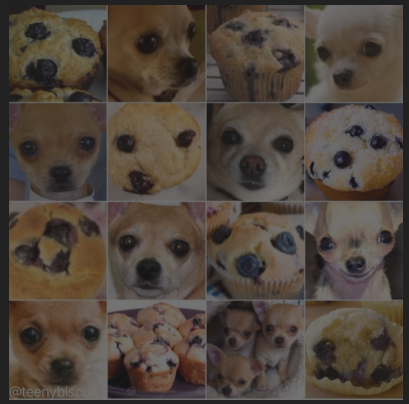
Loss



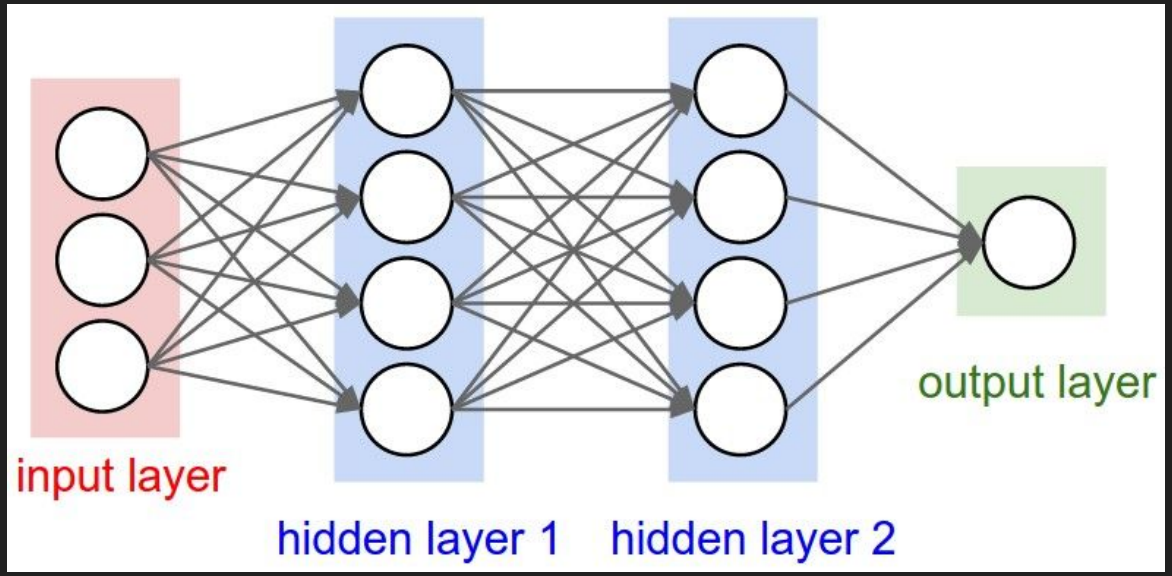
Optimizer



Data

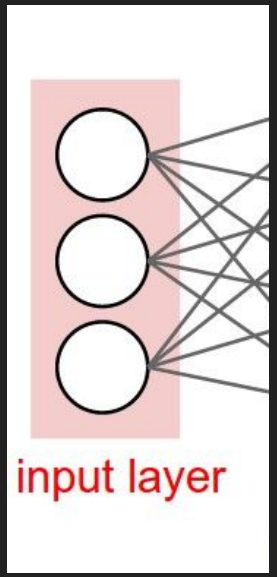


Fully Connected Neural Networks



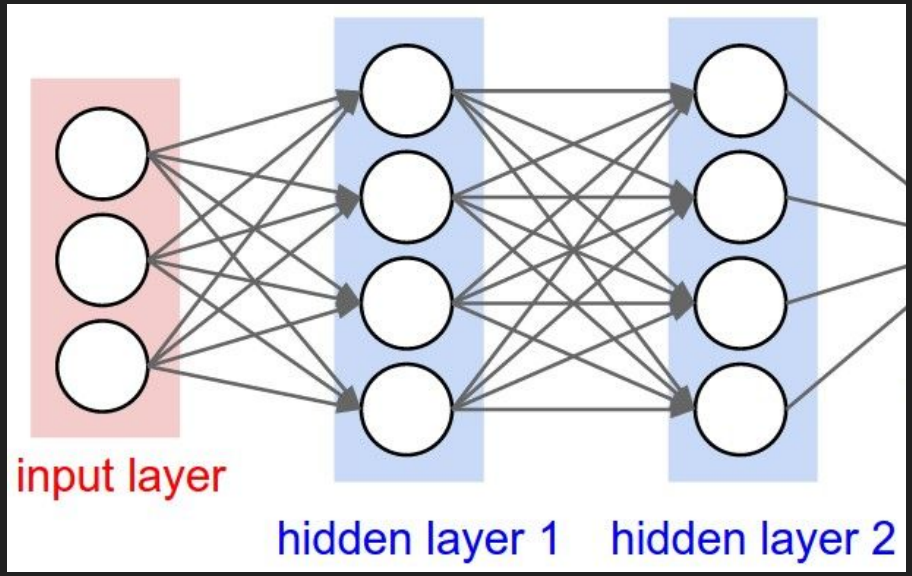
https://cs231n.github.io/

Input



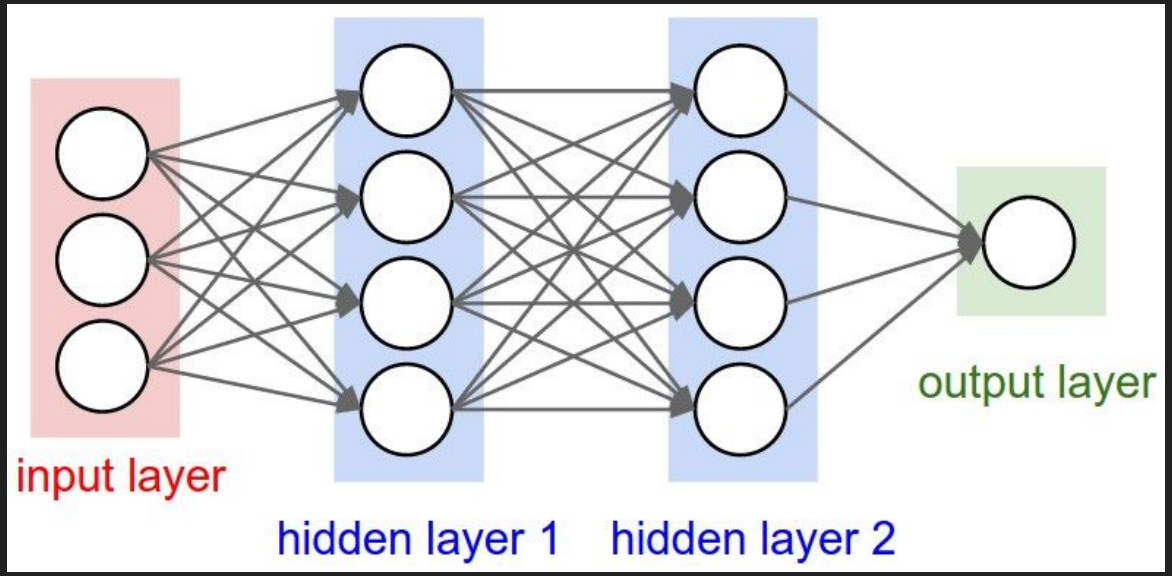
https://cs231n.github.io/

Hidden (intermediate) Layers



https://cs231n.github.io/

Output



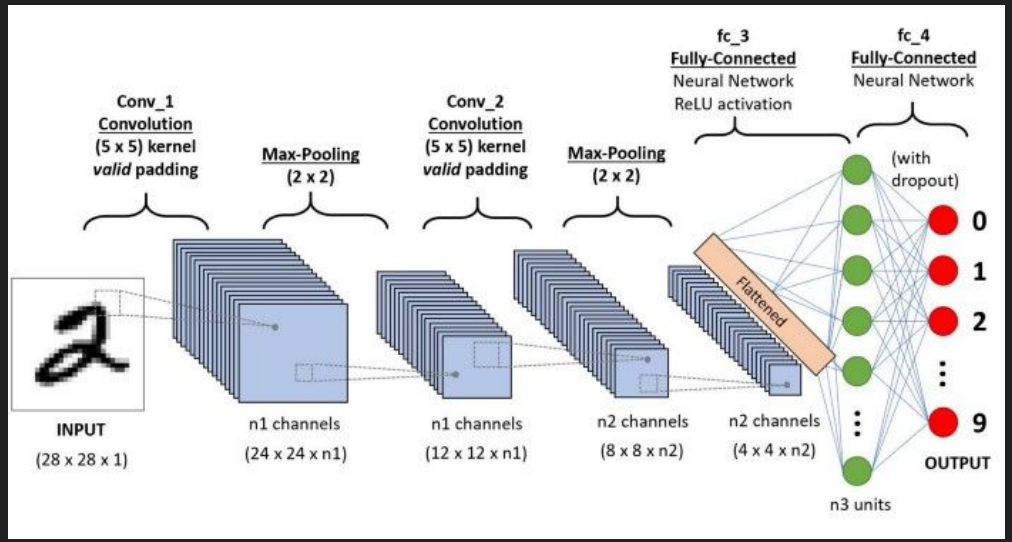
https://cs231n.github.io/

Model Assumptions

Why do we need different types of models?

Convolutional Neural Networks

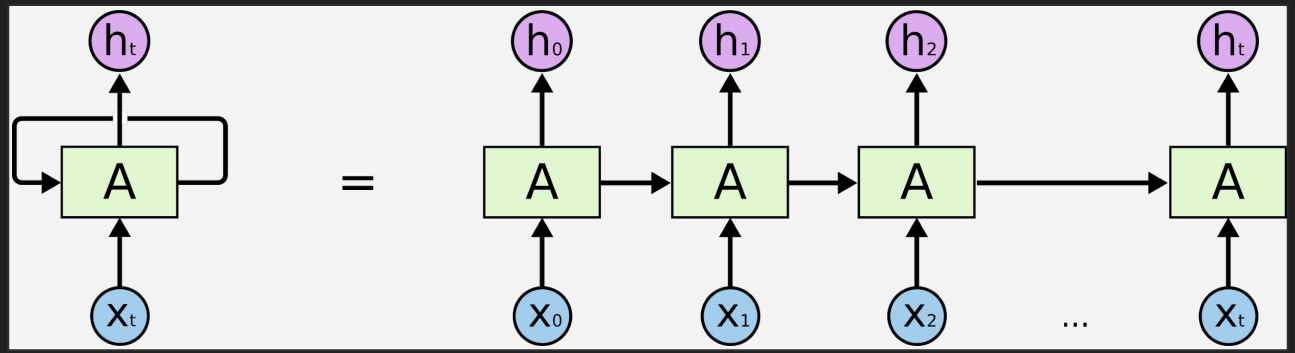
Assume inputs only affect other inputs if they're close together



https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Recurrent Neural Networks

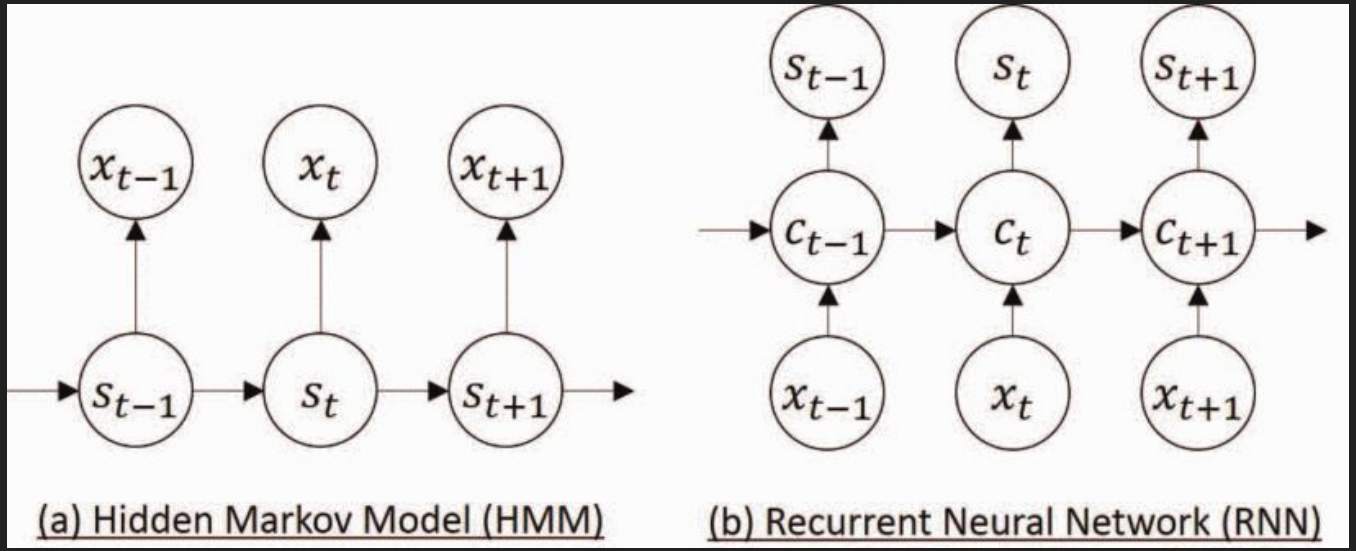
Assume order matters



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Recurrent Neural Networks

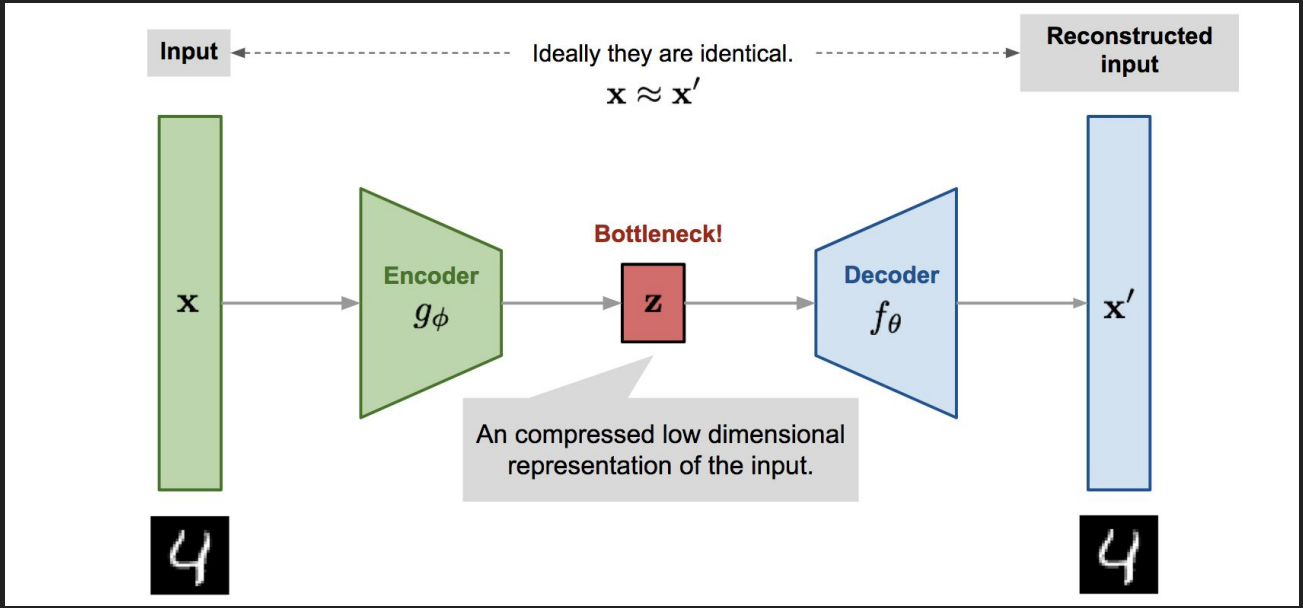
Assume order matters



https://ieeexplore.ieee.org/document/7404768

Autoencoders

Assume there is a lower dimensional space that the data lives on (the manifold hypothesis)



https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

Other Model Types

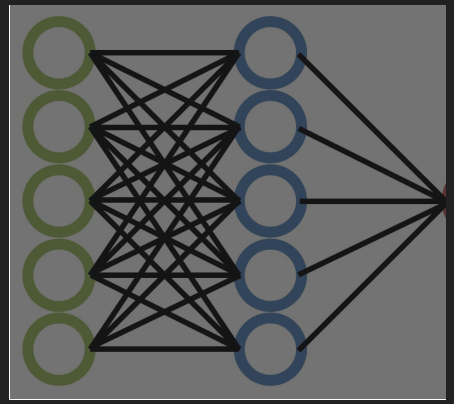
Generative Adversarial Networks (GANs) - Used in generating data

Graph Neural Networks - Used in chemistry and protein structure

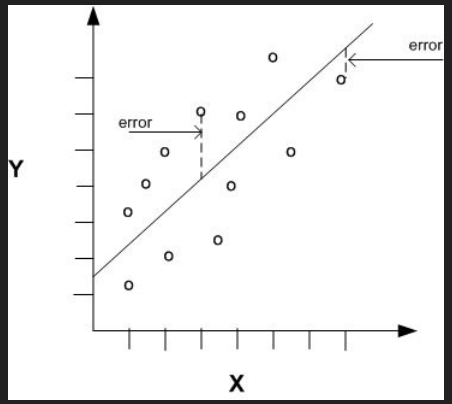
Transformers - Used for large text problems

Roadmap

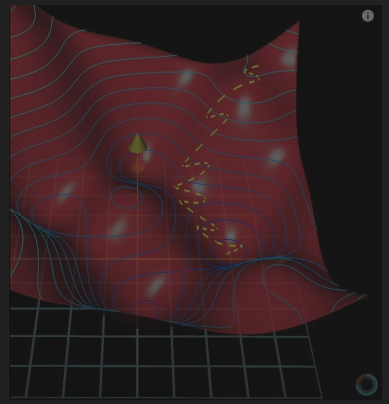
Model



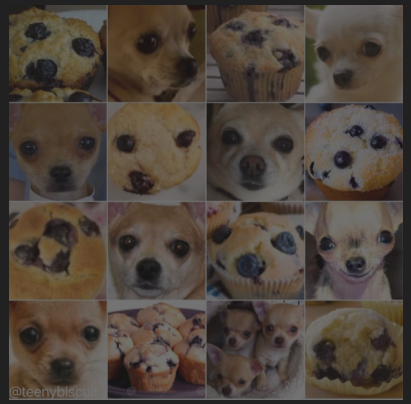
Loss



Optimizer



Data

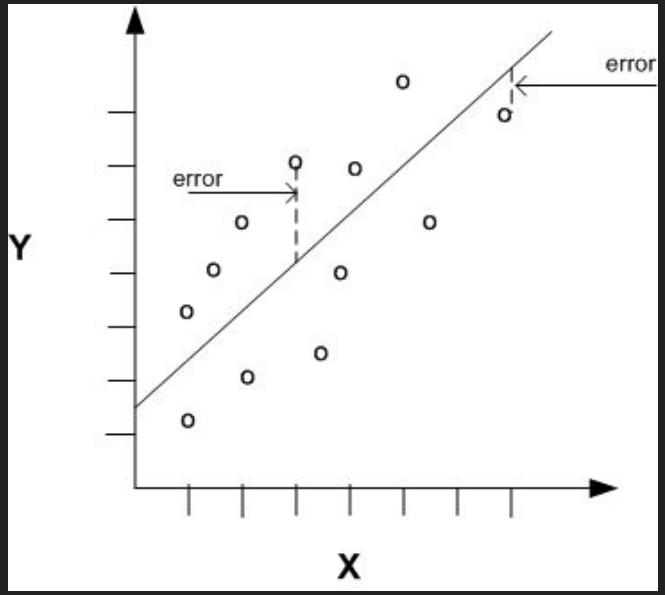


Loss

How right/wrong is each prediction?

Common Losses

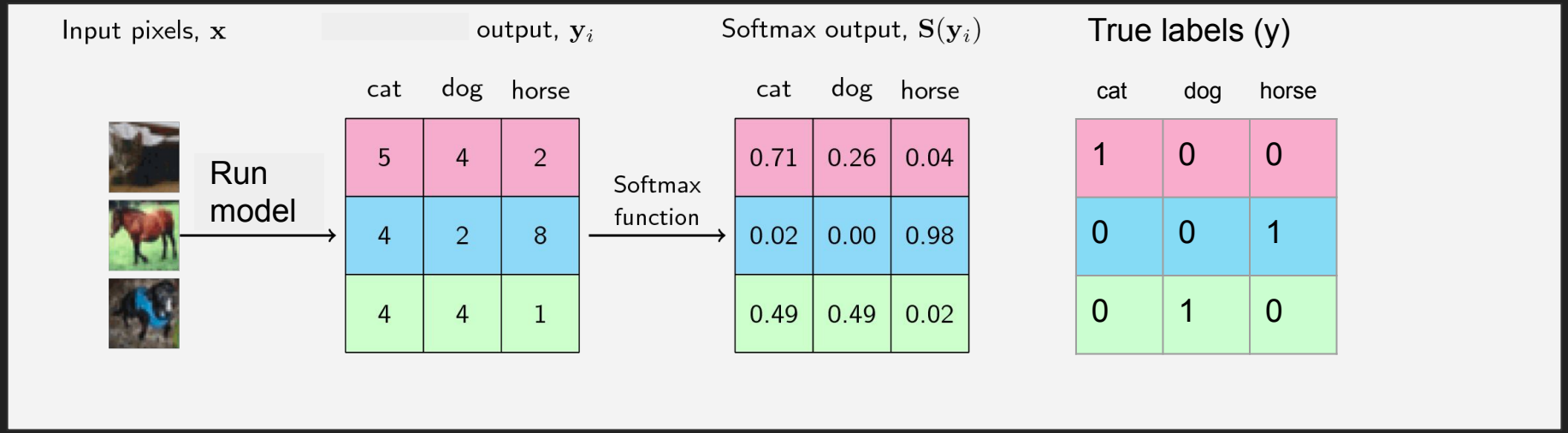
Mean Squared Error (MSE)



https://www.quora.com/What-is-the-meaning-of-root-mean-squared-error-RMSE-in-statistics

Common Losses

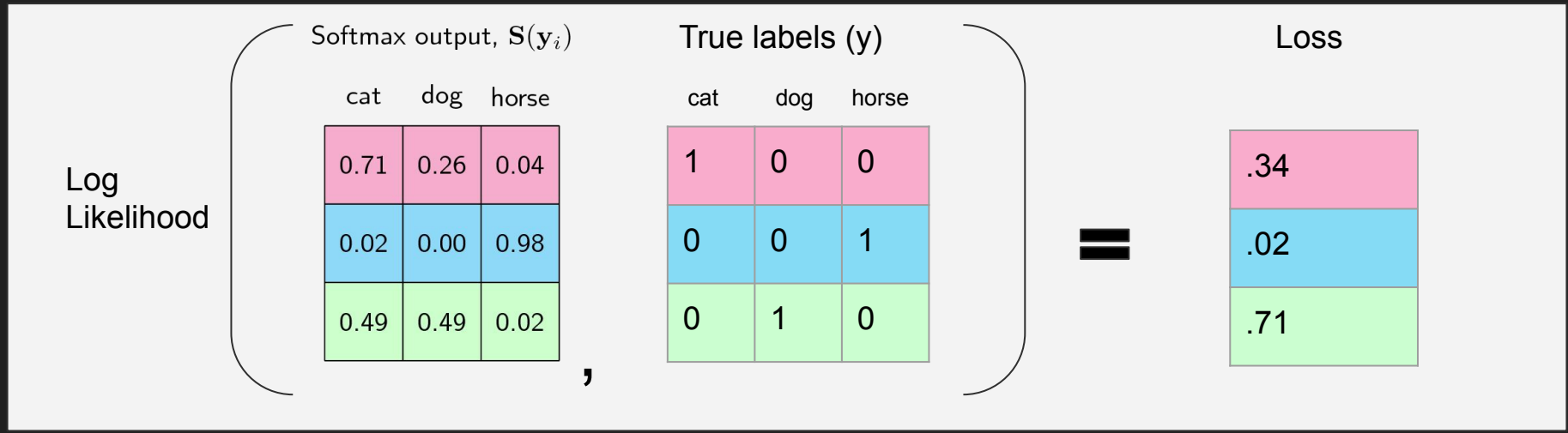
Negative Log Likelihood (Cross Entropy Loss)



ljvmiranda921.github.io

Common Losses

Negative Log Likelihood (Cross Entropy Loss)

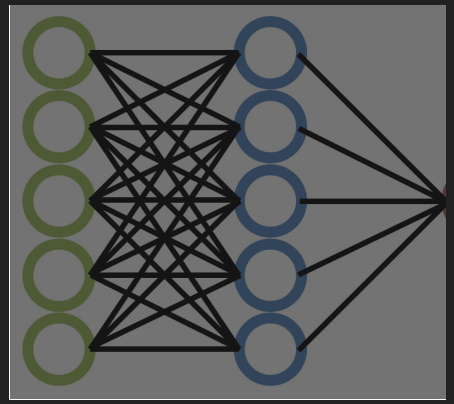


And More!

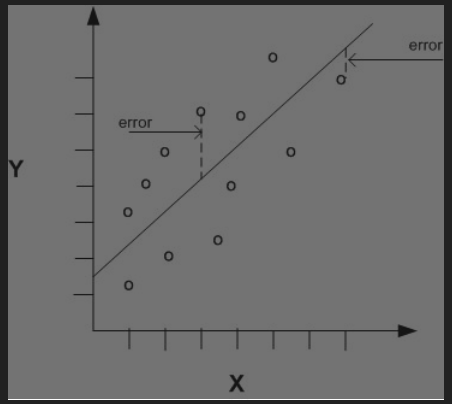
`nn.L1Loss`	Creates a criterion that measures the mean absolute error (MAE) between each element in the input x and target y .
`nn.MSELoss`	Creates a criterion that measures the mean squared error (squared L2 norm) between each element in the input x and target y .
`nn.CrossEntropyLoss`	This criterion combines `nn.LogSoftmax()` and `nn.NLLLoss()` in one single class.
`nn.CTCLoss`	The Connectionist Temporal Classification loss.
`nn.NLLLoss`	The negative log likelihood loss.
`nn.PoissonNLLLoss`	Negative log likelihood loss with Poisson distribution of target.
`nn.KLDivLoss`	The `Kullback-Leibler divergence`_ Loss
`nn.BCELoss`	Creates a criterion that measures the Binary Cross Entropy between the target and the output:

Roadmap

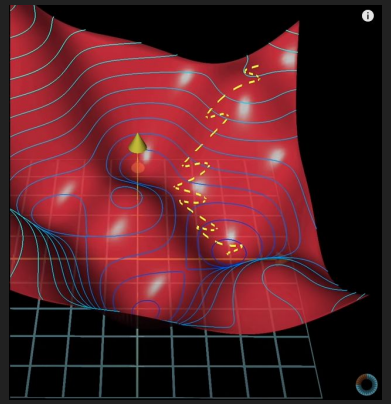
Model



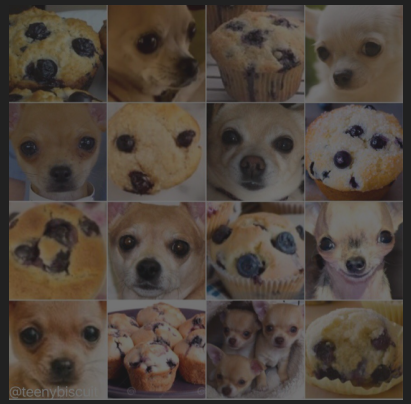
Loss



Optimizer



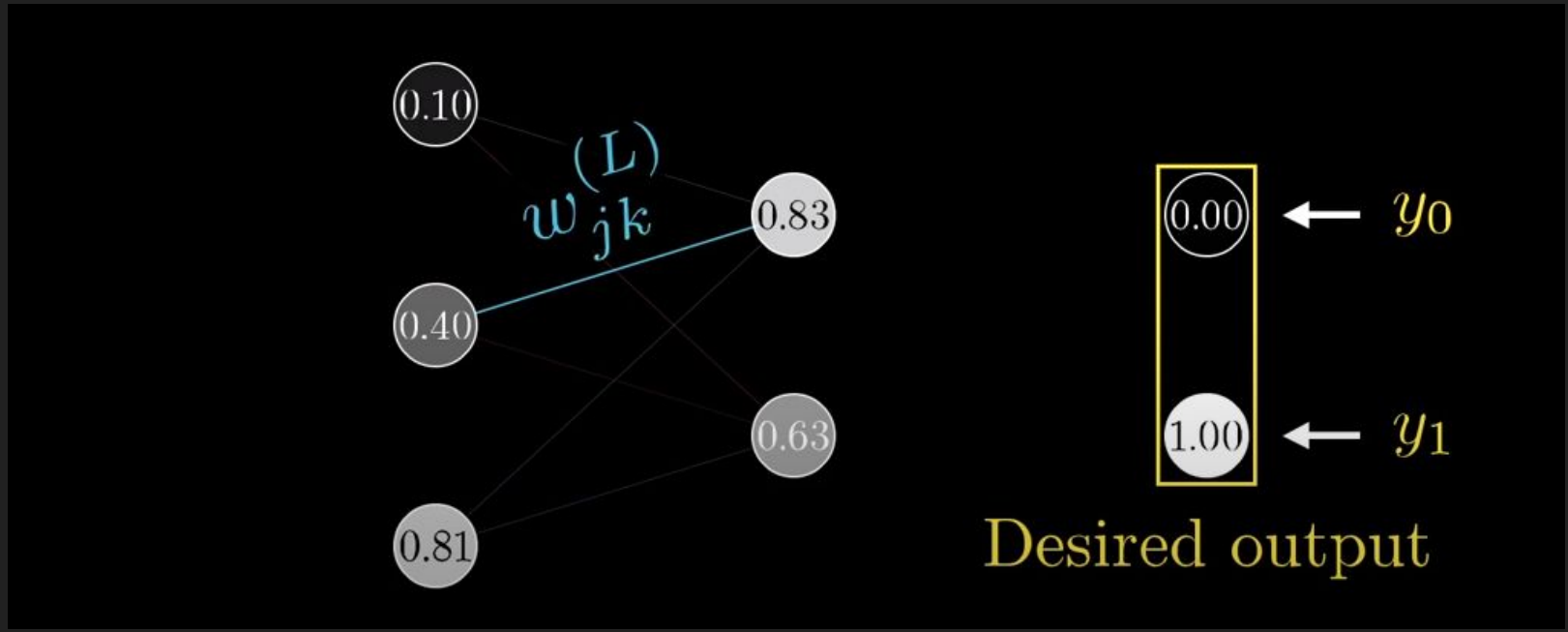
Data



Optimization

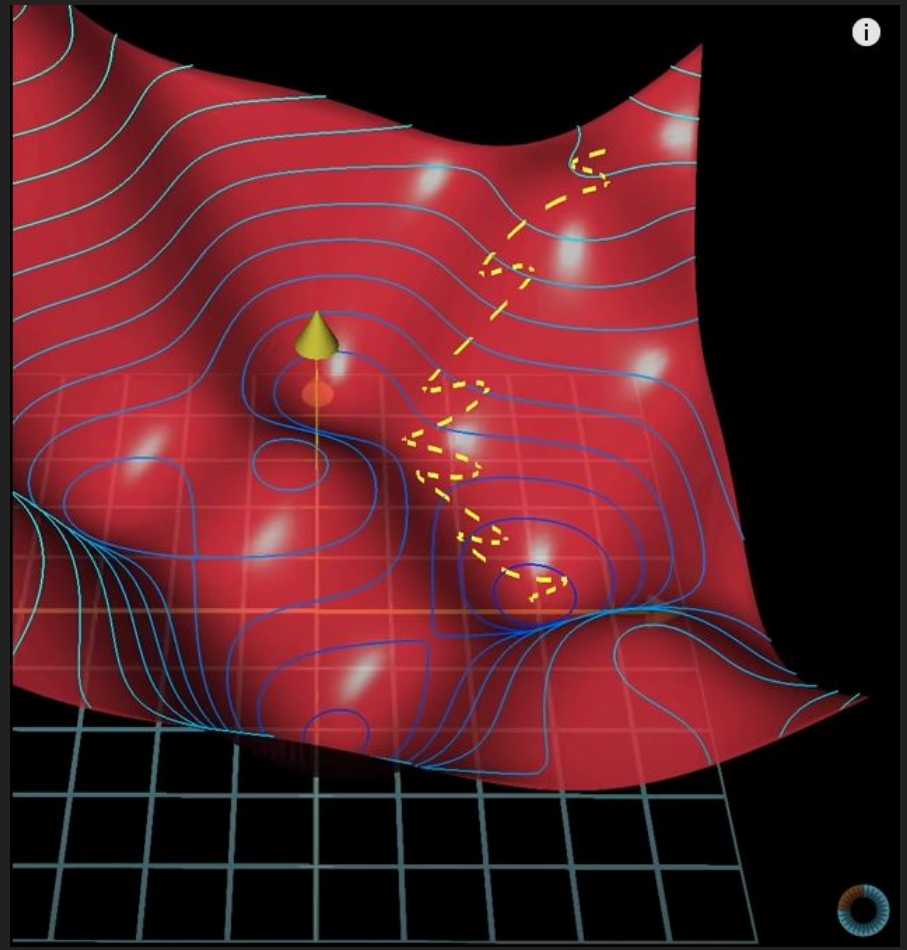
How do we pick what the weights should be?

Backpropagation



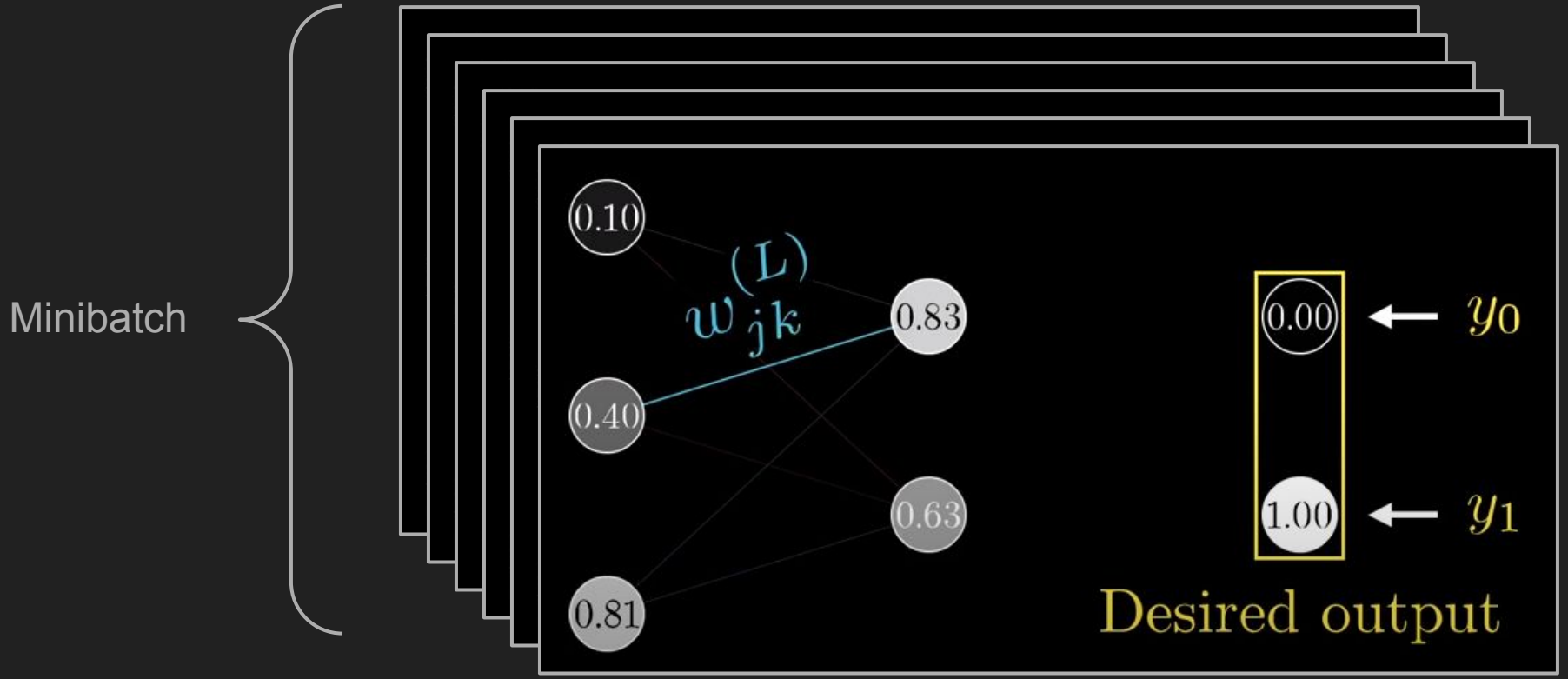
https://www.youtube.com/watch?v=tIeHLnjs5U8

Gradient Descent



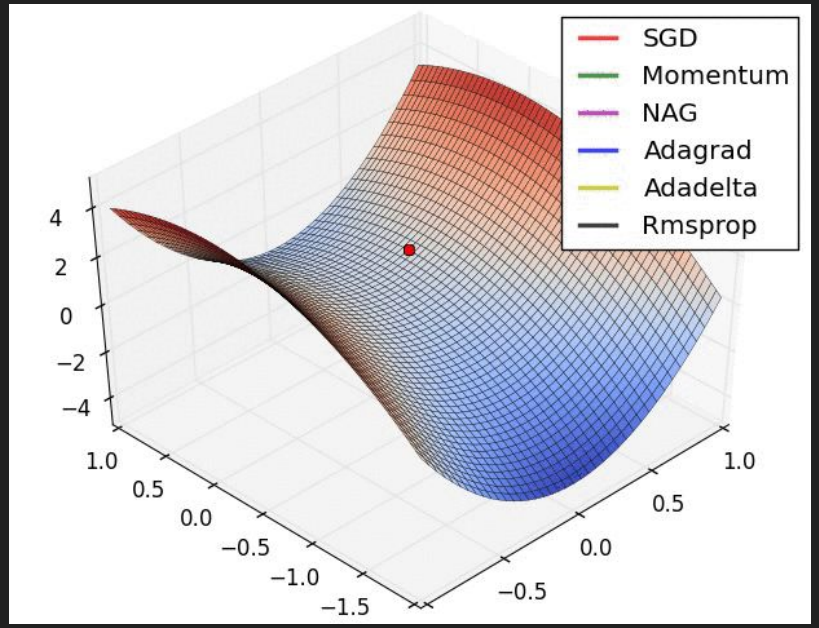
https://www.youtube.com/watch?v=Ilg3gGewQ5U

Gradient Descent



In Practice

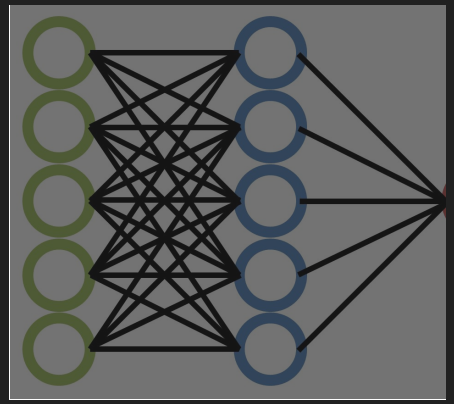
People typically use an adaptive optimizer like Adam



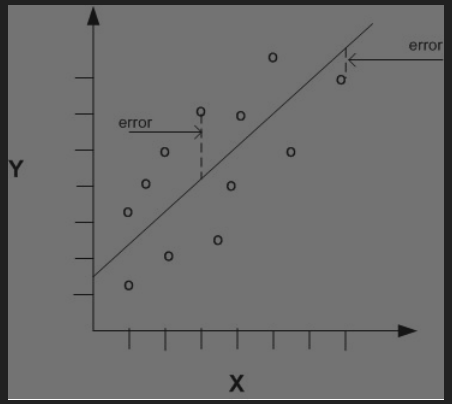
https://ruder.io/optimizing-gradient-descent/

Roadmap

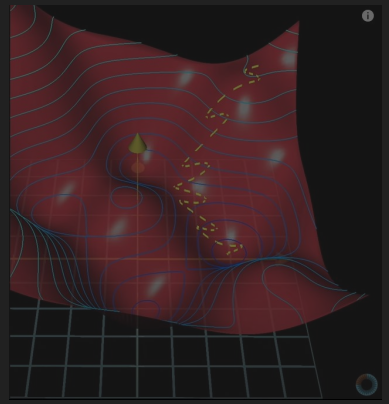
Model



Loss



Optimizer



Data

Data

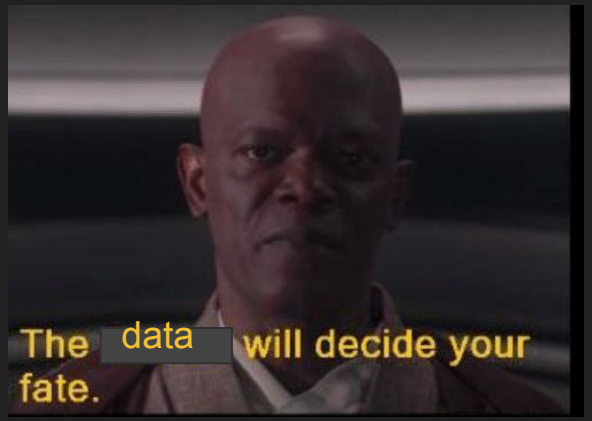


Photo cred: Star Wars

Data

Labeled:

X

https://twitter.com/teenybiscuit

y

[Muffin,
Dog,
Muffin,
Dog,
…,
Muffin]

Data

Unlabeled:

X

twitter.com/imgur

Data Splits

Train set - Used for training model

Tune set (Validation set) - Used for choosing which model/architecture to use

Test set - Used to see how well final model works

ML assumes the test set and the real world are independent and identically distributed

Data Augmentation

More data for free*!

https://bair.berkeley.edu/blog/2019/06/07/data_aug/

Data

Bio Considerations

Inter-study differences in labels/noise

Leakage between training and testing data

High base rates

Takeaways:

There are many uses for DL in Bio

Not all bio problems can/should be answered with DL

Pick a model that is appropriate for your problem

Questions?

Demo: Example Neural Network

Intermission (~ 5 minutes)