Reading 5

★ Video: Regularization Part 1: Ridge (L2) Regression

- A least squares fitted line is the line that results in the minimum sum of squared residuals.

- If you were to fit a line onto 2 points (lets say this is our training data) then the sum of squared residuals will be 0.
  - ↳ But when you introduce the testing data, the sum of squared residuals may increase rapidly.
    - ↳ If this were the case, then the new line is said to have high variance.
    - ↳ Basically, the new line is overfit to the training data.

- The main idea of ridge regression is to find a new line that doesn't fit the training data as well.
  - ↳ In other words, we introduce a small amount of Bias into how the New Line is fit to the data.
  - ↳ But in return for that small amount of Bias, we get a significant drop in variance.

- Starting with a slightly worse fit, Ridge Regression can provide better long term predictions.

★ • When Ridge Regression determines the values of the parameters in the equation of the line (y-intercept and slope) it minimizes the sum of the squared residuals + $[\lambda \cdot (\text{slope})^2]$

  - λ: determines how severe the penalty is
  - $(\text{slope})^2$: adds a penalty to the traditional least squares equation

- In our example the ridge regression line had a smaller slope than the original line.
  - ↳ This means the ridge regression line was less sensitive to changes in weight (x-axis).

- λ can be any value from $[0, \infty)$
  - ↳ if $\lambda = 0$ then the ridge regression line equals the original line.
  - ↳ As you increase your value for λ, the slope gets asymptotically closer to 0.

- To pick a value for λ, you test a lot of λ values and use 10-fold Cross Validation to determine which one results in the lowest variance.

- Ridge Regression also works for discrete data instead of continuous data.
  - ↳ Ex Normal Diet vs High Fat diet (comparing against size)
  - ↳ first draw a line at the mean size for each group
  - ↳ the residuals are how far a data point is from its respective mean line.

- Least Squares still minimizes the sum of the residuals squared.
- Ridge Regression minimizes:

  Sum of squared residuals + (the distance between the two mean lines)^2

- When λ increases, the ridge regression line becomes less sensitive to the difference of the means of normal diet and high fat diet.

- Ridge Regression can be used in logistic regression as well.

- Ridge Regression can be applied to more complex situations as well.
  - In general the ridge regression penalty contains all of the parameters except for the y-intercept.
    - so it be $\lambda \cdot [(\text{factor1})^2 + (\text{factor2})^2 + (\text{factor3})^2 + \dots]$

- With only one data point we cannot find a least squares line of best fit.
  - If you have 3 parameters, you cannot find a least squares line of best fit without at least 3 data points.
  - Basically for a problem with 10,000 parameters, you need <u>at least</u> 10,000 data points to solve for all the parameters in a least squares line.

- But ridge regression <u>can</u> find a line with less data points than parameters.

- <u>NOTE</u>: Ridge regression is especially handy when you do not have a lot of training data.

☆ Video! Regularization Part 2: Lasso (L1) Regression

- Lasso regression is very similar to ridge regression, but it has some key and important differences.

- Ridge Regression minimizes:
  - ↳ Sum of squared residuals + $\lambda \cdot (\text{slope})^2$ (for 2 dimensions)

☆ - Lasso Regression minimizes:
  - ↳ Sum of squared residuals + $\lambda \cdot |\text{slope}|$ (for 2 dimensions)

- Both ridge and lasso regression can be applied in the same contexts.
- Lasso regression also doesn't include the y-intercept in the penalty

☆ - The big difference between ridge regression and lasso regression is that ridge regression can only:
  - ↳ shrink the the slope asymptotically close to 0.

while lasso regression:
  - ↳ can shrink the slope all the way to 0.

☆ - If you had an output predicted by 2 good parameters and 2 bad parameters:
  - ↳ lasso regression would allow you to shrink the influence of the bad parameters all the way down to 0.
  - ↳ In other words, you would only predict the output based on the two good parameters.

☆ - In contrast, Ridge Regression tends to do a little better when most parameters are useful.

✱ Video Welch Labs: Perceptron

- Perceptrons are the atomic unit of modern AI.

- To learn, you feed training example and turn all the corresponding dials to right by a constant value called the learning rate.
  - ↳ You turn all other dials to the left by the learning rate.

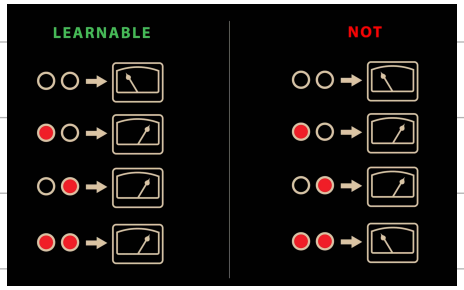
- You keep feeding examples and adjusting weights of dials until consistent output prediction is reached.

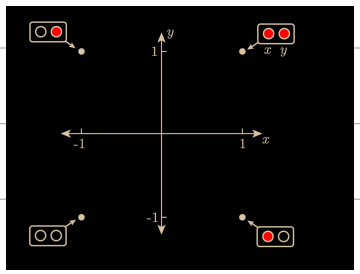
- This is the perceptron learning rule.

- The perceptron can easily tell apart patterns.

- You can introduce a bias value (dial) that directly allows you to add or subtract from the final value.
  - ↳ The bias dial is not connected to any of the inputs.

- Perceptron learnable vs not learnable:





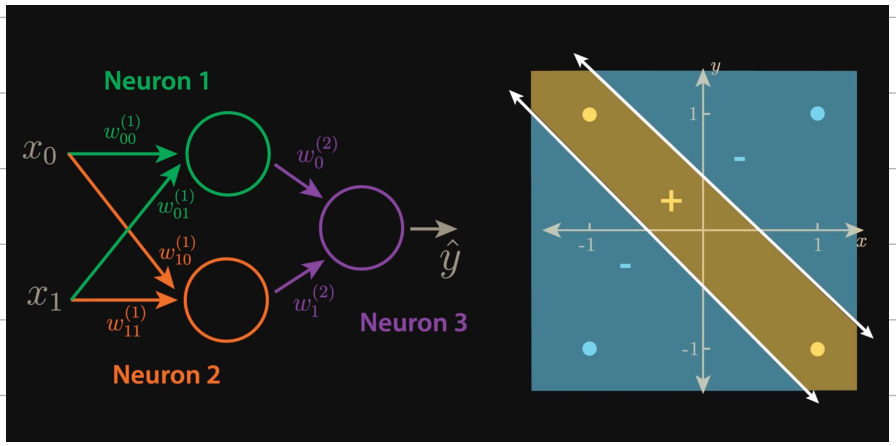
↳ inputs displayed mathematically

- the output is : $\text{Output} = w_1 x + w_2 y + b$
  - w_1, w_2: weights of first & second dial
  - b: weight of bias value

- the perceptron learns the line that differentiates the good outcomes from bad outcomes.

- so the perceptron can only produce reliable outputs if the data is linearly seperable.

- There is a way to fix this though.
  - ↳ You have to add more neurons : A neuron is a perceptron



- The perceptron learning rule for a single perceptron does not generalize well to multi-layer perceptrons.

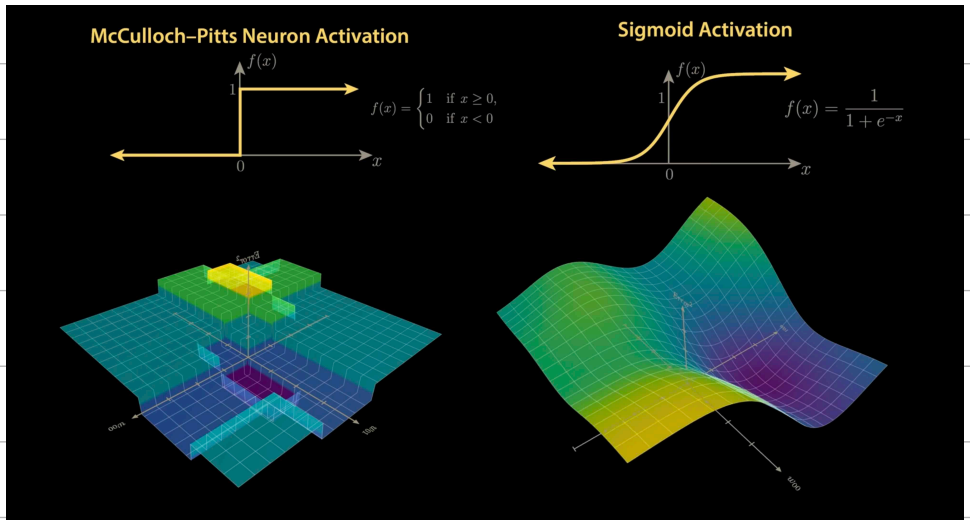
- So it was hard to actually find the proper weights.

- Error is the difference between machine (perceptron) output and the desired output

- When you mess around with the dials to minimize the error, you see that the plot of $(error)^2$ forms a bowl in 3D.
  - ↳ So you use the gradient to continuously step down until you find the local minimum.

- This was called the least mean squares (LMS) algorithm.

- The original perceptron assumed 0 or 1 output (either on or off)
  - ↳ This made the error function have cliffs of infinite steepness so you couldn't gradually walk down to the minimum.
  - ↳ This is why the sigmoid function was introduced to flatten out the error function, making it possible to use gradient descent.



- Modern systems use backpropogation algorithms to train.

- Chat GPT3 uses an chain of 96 layers:
  - ↳ each layer consists of an Attention block and a MLP block:
    - ↳ MLP block:
      - ↳ has 2 layers of perceptrons
        - ↳ 1st layer has ~ 50,000 neurons
        - ↳ 2nd layer has ~ 12,000 neurons
    - ↳ Attention block:
      - ↳ essentially a specialized MLP where the weights are controlled by other perceptrons.
        - ↳ This allows the data itself to move the weights
      - ↳ each attention block uses ~ 50,000 neurons.

- So GPT3 has ~ 10 million neurons in total