Linear Models

Forest Agostinelli
University of South Carolina

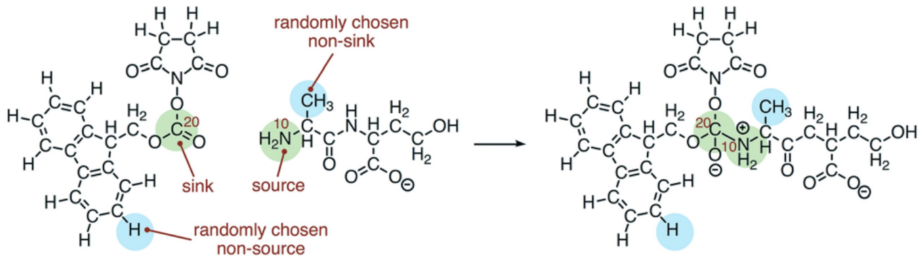
Outline

- Supervised Learning
- Linear regression
- Logistic regression
- Explainability
- Hypothesis space

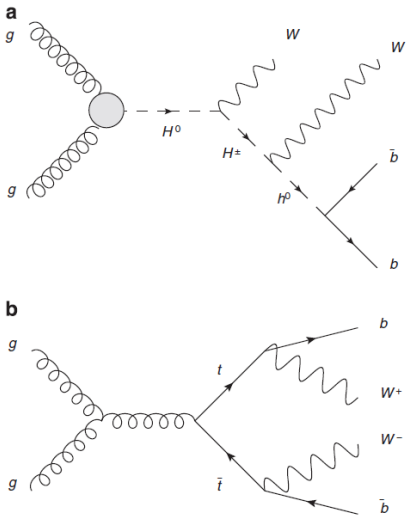
Supervised Learning

- We would like to find a hypothesis to explain a set of observations
- Each observation is an input and output pair
  - Medical images -> disease classification
  - Protein -> 3D structure
  - Chemical reactants -> chemical products
- We may need to consider many different hypotheses to find a good explanation
- We still may not find a hypothesis that explains all the observations
- The types of hypotheses that we can learn are determined by the hypothesis space

Supervised Learning



Chemical reaction prediction



Exotic particle detection

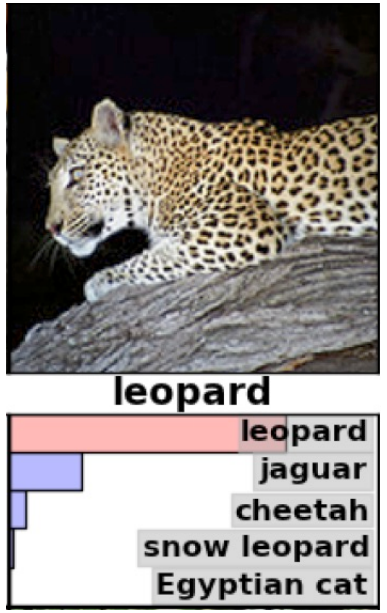
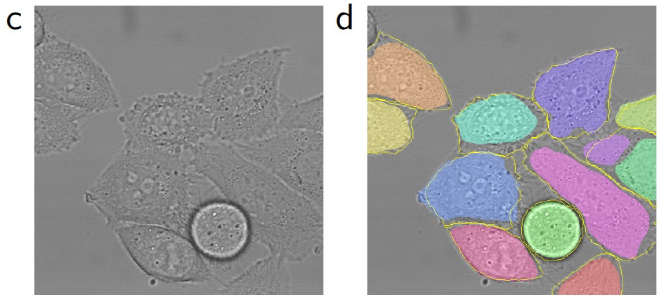
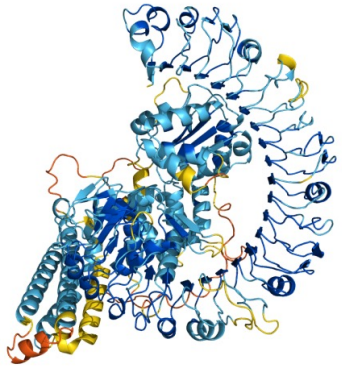


Image classification



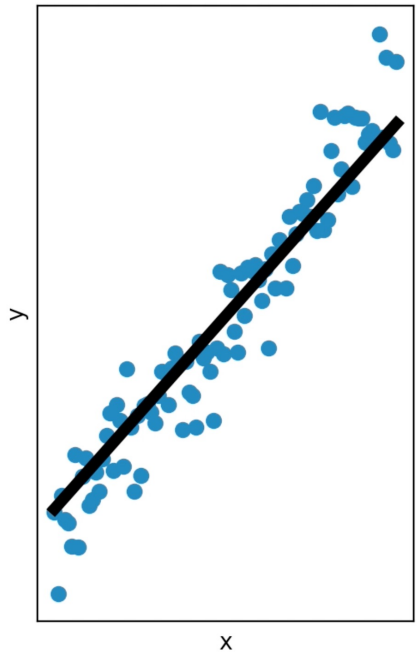
Medical image segmentation



Protein folding

Regression and Classification

- Find $\boldsymbol{\theta}$ that best maps the input $\boldsymbol{x} \in \mathbb{R}^p$ to output $\boldsymbol{y} \in \mathbb{R}^q$
  - $\boldsymbol{y} = f(\boldsymbol{x}, \boldsymbol{\theta})$
- The input $\boldsymbol{x}$ is also known as the features or predictors
- $f(\boldsymbol{x}, \boldsymbol{\theta})$ is also known as the model
- **Regression**: $\boldsymbol{y}$ is a continuous variable
- **Classification**: $\boldsymbol{y}$ is a categorical variable



Training a Model

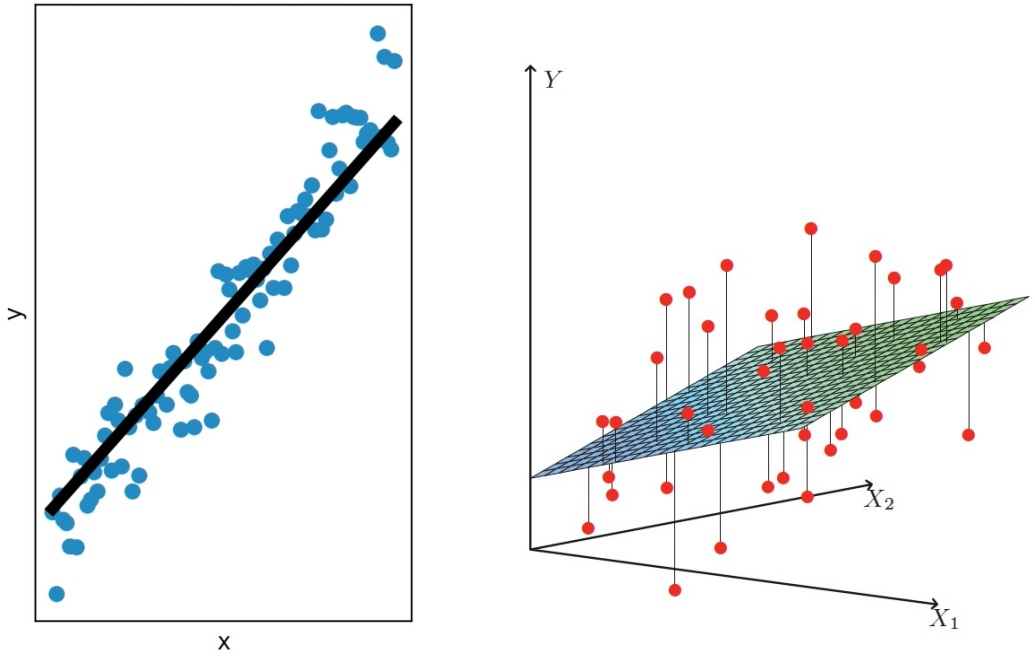
- Model: A function that maps inputs to outputs
  - $\boldsymbol{y} = f(\boldsymbol{x}, \boldsymbol{\theta})$
- Parameters ($\boldsymbol{\theta}$): Determines this mapping
- Dataset: A set of examples $\{\dots, (\boldsymbol{x_i}, \boldsymbol{y_i}), \dots\}$
- Loss function: Measures performance based on the parameters and the dataset
- We then seek to find parameters that optimize our given loss function

Outline

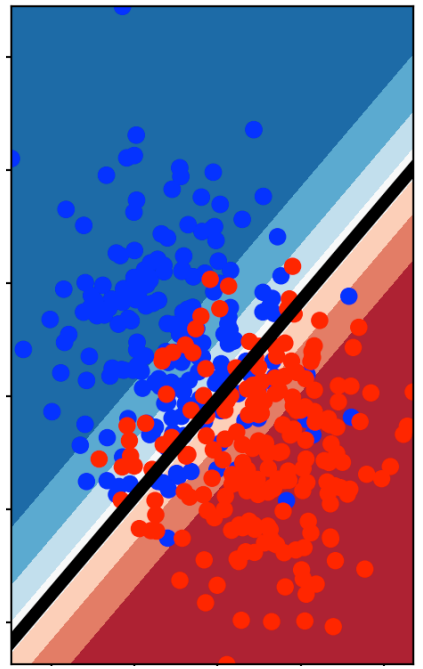
- Supervised Learning
- Linear regression
- Logistic regression
- Explainability
- Hypothesis space

Linear Models

- Limits mapping of inputs to outputs to a line
- Regression: linear function
- Classification: linear decision boundary



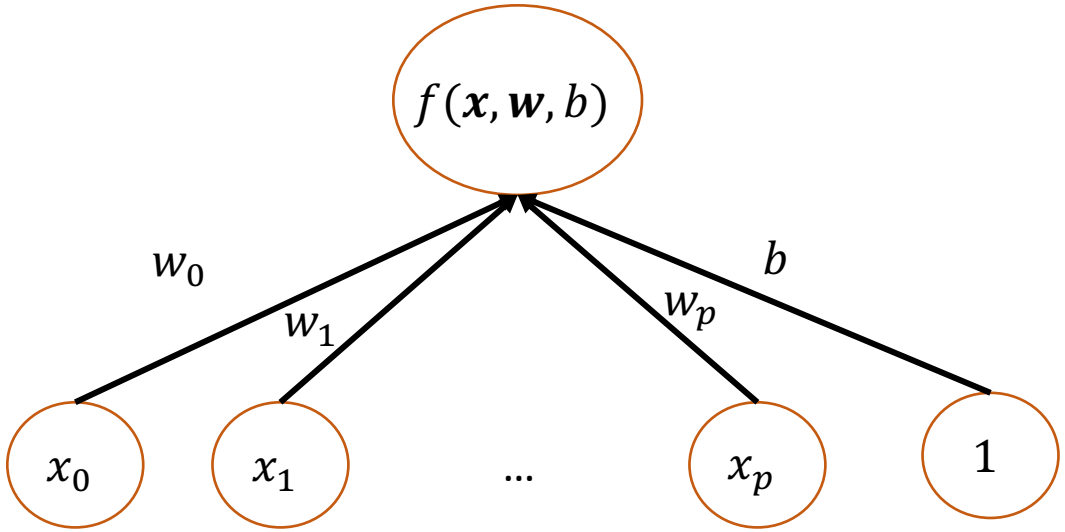
Regression



Classification

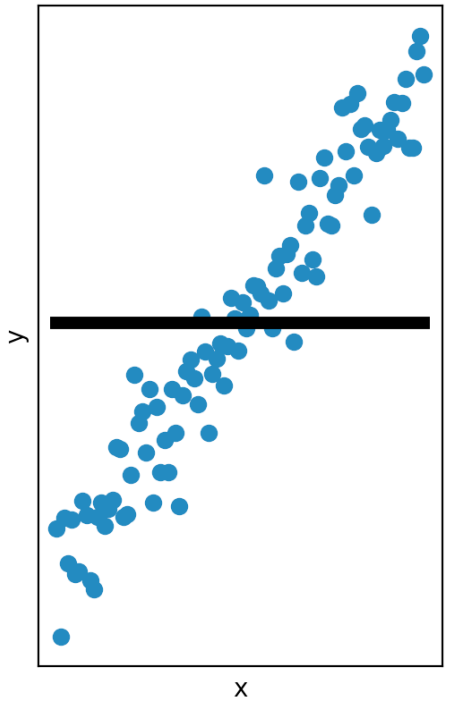
Linear Regression

- Learning a function $f(\boldsymbol{x}, \boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$
  - Linear model $\boldsymbol{\theta} = [\boldsymbol{w}, b]$
- $f(\boldsymbol{x}, \boldsymbol{w}, b) = \boldsymbol{w}^T \boldsymbol{x} + b = \sum_i w_i x_i + b = w_0 x_0 + w_1 x_1 + \cdots + w_p x_p + b$



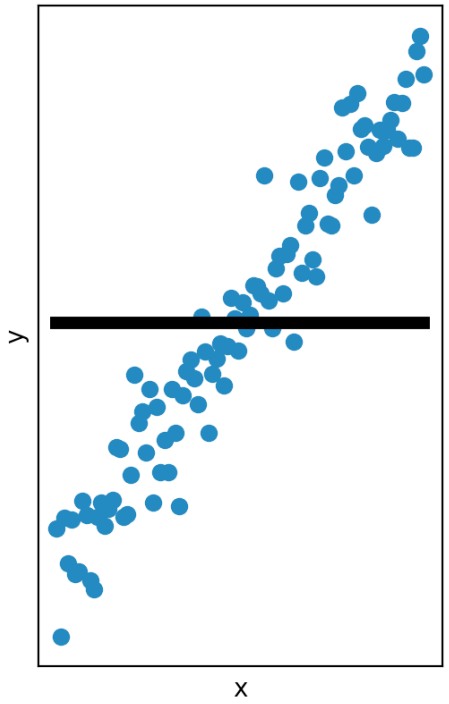
Linear Regression: Example

- Imagine we are trying to predict the yield of a chemical reaction (y-axis) as a function of temperature (x-axis)
- Dataset: we measure the yield at a 100 different temperatures and wish to learn a model that predicts yield at ***all*** temperatures
- Our linear model has a single parameter, w, that determines the slope of the line
  - $f(x, w) = wx$



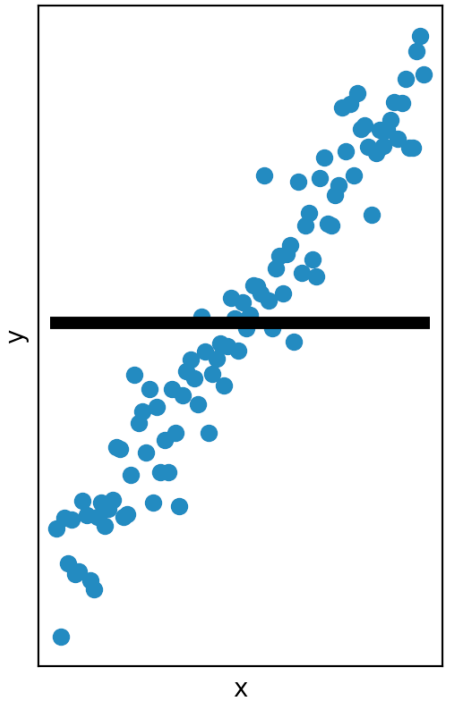
Linear Regression: Loss Function

- Given the following data and linear model, what is a good loss function to determine how well our model fits the data?



Linear Regression: Loss Function

- Given the following data, linear model, and loss function how do we find the best parameters that minimizes the loss function?
- $\mathcal{L}(w) = \frac{1}{2n}\sum_n (y_n - wx_n)^2$
  - Mean squared error

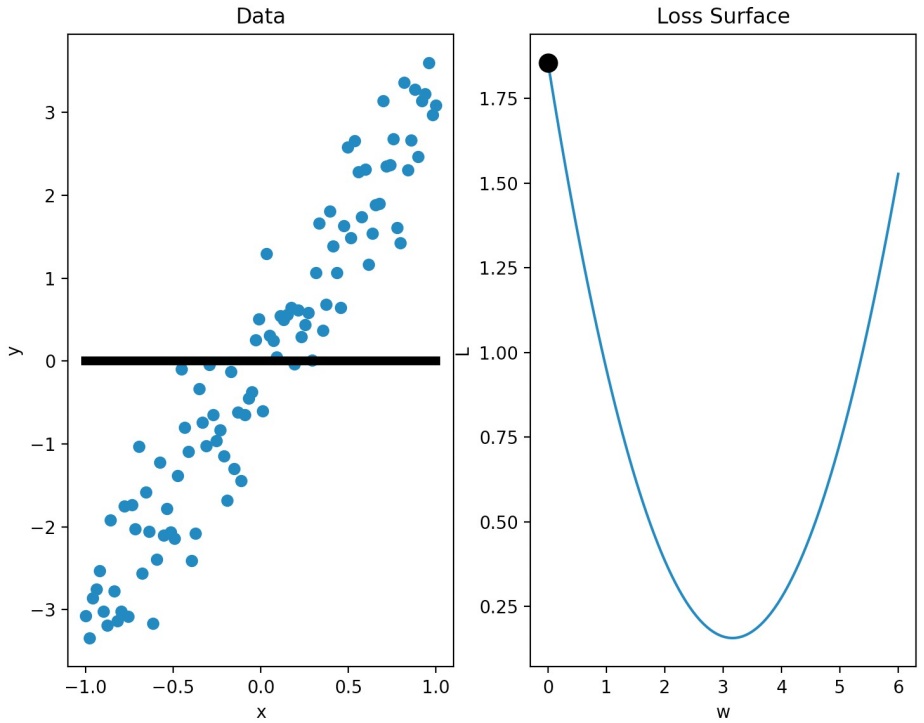


Optimization: Analytical Solution

- $\mathcal{L}(\boldsymbol{\theta}) = \sum_n \big(y_n - f(\boldsymbol{x}, \boldsymbol{\theta})\big)^2 = ||\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}||_2^2$
- $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = 2\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) = 0$
- $\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
- Not possible for deep neural networks
- We will focus on other solutions to build our intuition for deep neural networks

Optimization: Loss Landscape

- Since we only have one parameter, w, we can plot the loss as a function of the parameter and see if we can visually observe when the loss is minimized
- However, since we can only see in three dimensions, as the number of parameters grows, visualizing the loss surface becomes more difficult
- How can we find a general algorithm for finding a set of parameters that minimizes our loss function?

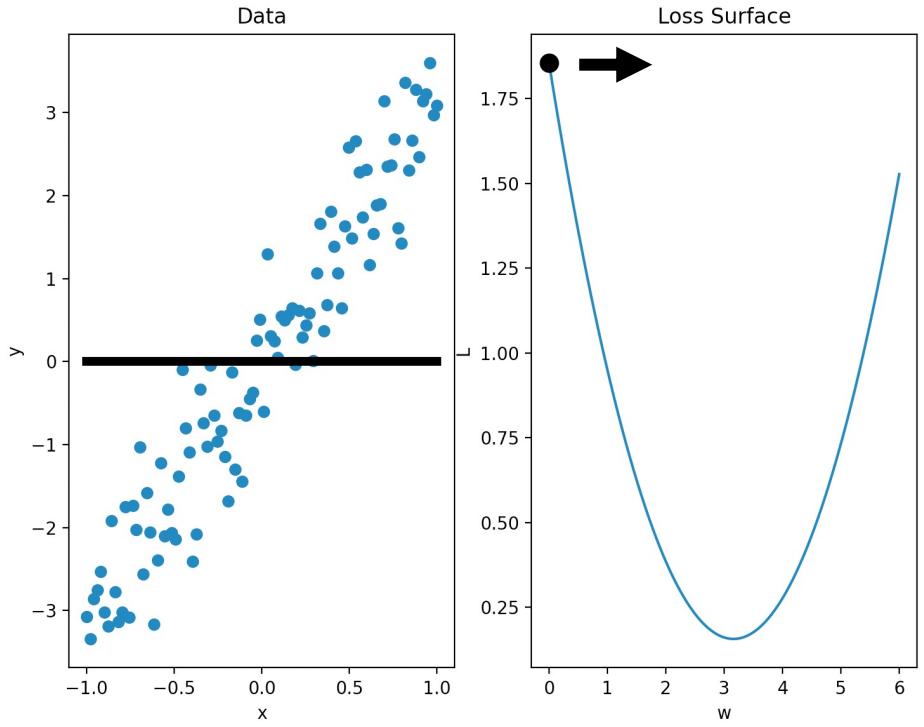


Optimization: Other Methods

- Random search
  - Try K random values and pick the best one
- Line search
  - Uniformly pick K values in between a set min and set max value and pick the best one
- Evolutionary methods
  - Use the loss function as a measure of “fitness” to determine what parameter values get to be combined to produce new parameter values

Optimization: Gradient Descent

- While, in higher dimensions, we cannot visualize the loss landscape, using calculus, we can still determine the direction of steepest descent and the rate at which the loss function is decreasing
- In this example, we see that by moving w in the positive direction the loss function decreases up until a certain point
- However, in general, given the direction and rate of change, we still do not know *how much* we should move in that direction



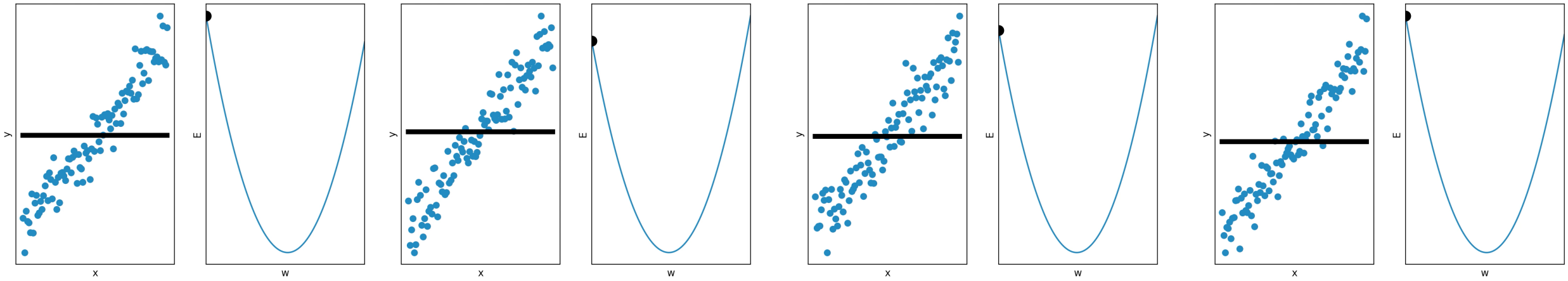
Gradient Descent: Algorithm

- Get dataset, D
- Initialize parameters, $\boldsymbol{\theta}$
- For number of training steps
  - $\mathcal{L}(\boldsymbol{\theta}, D)$ # calculate loss
  - $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})$ #calculate gradient
  - Move in opposite direction of the gradient to adjust parameters

- How much should we adjust $\boldsymbol{\theta}$ based on the gradient?

Gradient Descent: Learning Rate

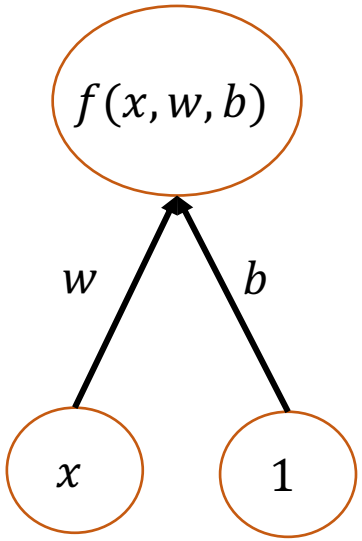
- The learning rate, α, determines how large of a step we should take
  - i.e. the larger the learning rate, the larger the step size
- The step size is proportional to the gradient
  - i.e. the larger the gradient, the larger the step size
- Which learning rate is best?



$\alpha = 1.0$ $\alpha = 0.1$ $\alpha = 5.0$ $\alpha = 10.0$

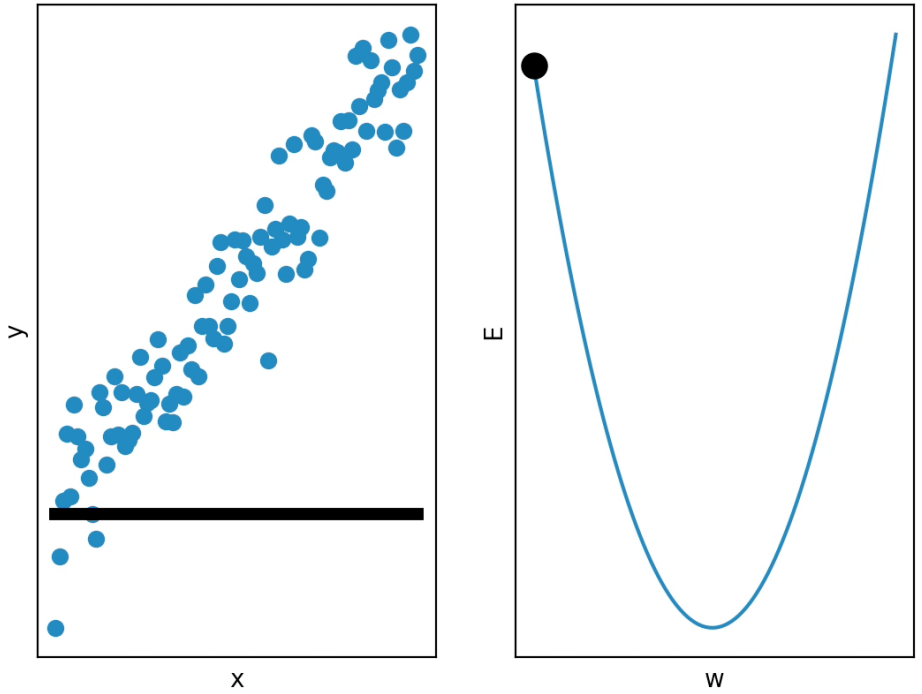
Linear Regression: 1D with Bias

- Recall the general form of linear regression
  - $f(\boldsymbol{x}, \boldsymbol{w}, b) = \boldsymbol{w}^T \boldsymbol{x} + b = \sum_i w_i x_i + b = w_0 x_0 + w_1 x_1 + \cdots + w_p x_p + b$
- For a one-dimensional input, this becomes
  - $f(x, w, b) = wx + b$
- A bias lets us shift the function up and down



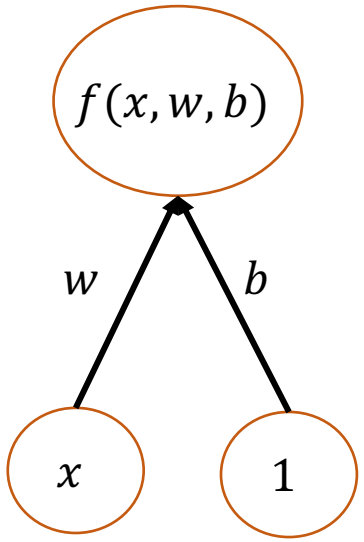
Linear Regression: 1D with Bias

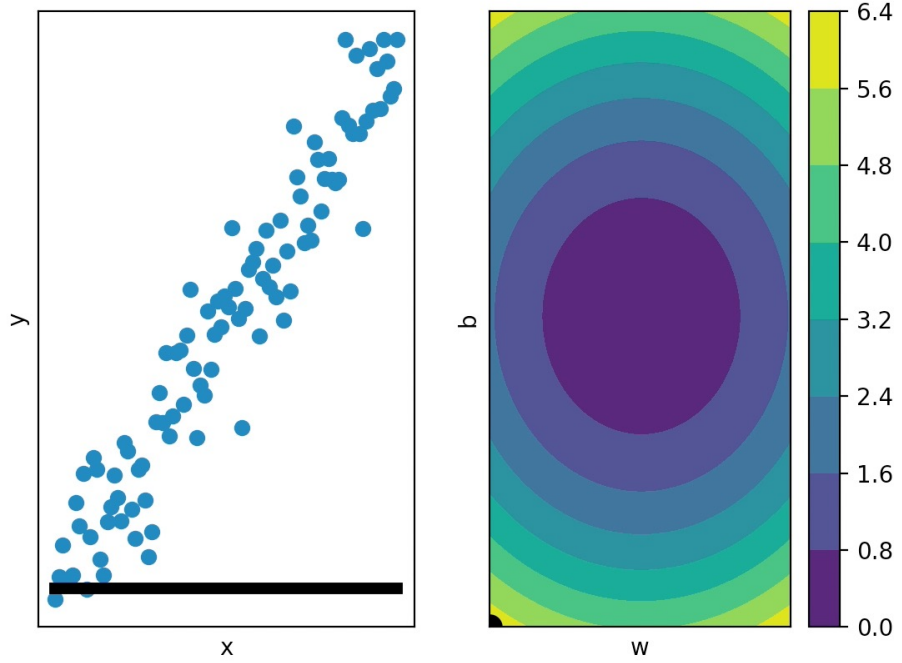
- While we may minimize our loss function, we may still learn a poor function
- For example, linear regression with no bias when the training data is shifted up



Linear Regression: 1D with Bias

- With a bias parameter we now have two parameters
- The gradient descent algorithm remains the same

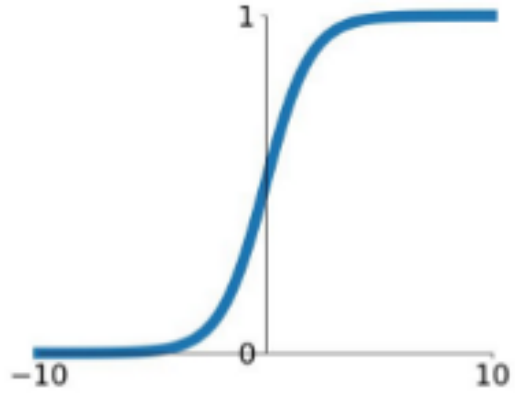




Bias $\alpha = 0.5$

Binary Classification

- We would like to differentiate between 2 classes
  - Dog/cat
  - Disease/no disease
  - Pedestrian/no pedestrian
- We are given an input vector $\boldsymbol{x}$ and want to predict y
- We can use a logistic function to bound the output of our model between 0 and 1 to represent the probability that it is one class or the other
  - $\sigma(a) = \frac{1}{1+e^{-a}}$
- Hence the name, logistic regression

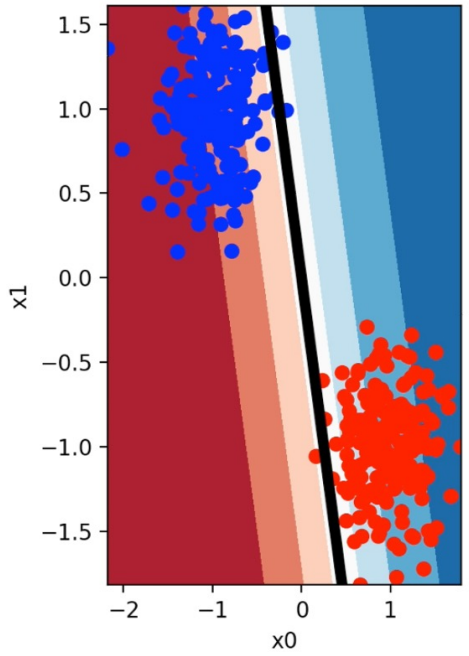


Outline

- Supervised Learning
- Linear regression
- Logistic regression
- Explainability
- Hypothesis space

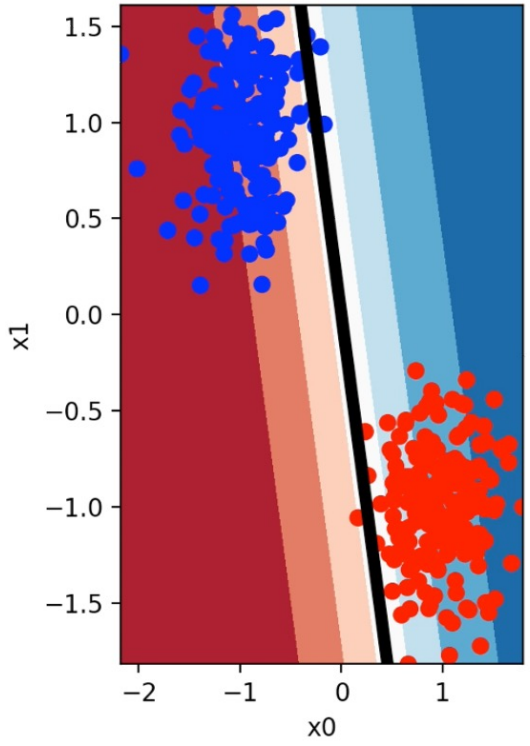
Logistic Regression: Example

- Imagine two measurements from medical tests we would like to predict the presence of a disease
- Dataset: we do the two medical tests on 100 different patients on whom we know with high confidence whether or not they have the disease (e.g. because we did a more invasive procedure to determine its presence) and wish to learn a model that the presence of this disease on **any** patient
- Our linear model has two parameters, w_0 and w_1, that determine the decision boundary
  - $f(\boldsymbol{x}, \boldsymbol{w}) = w_0 x_0 + w_1 x_1$



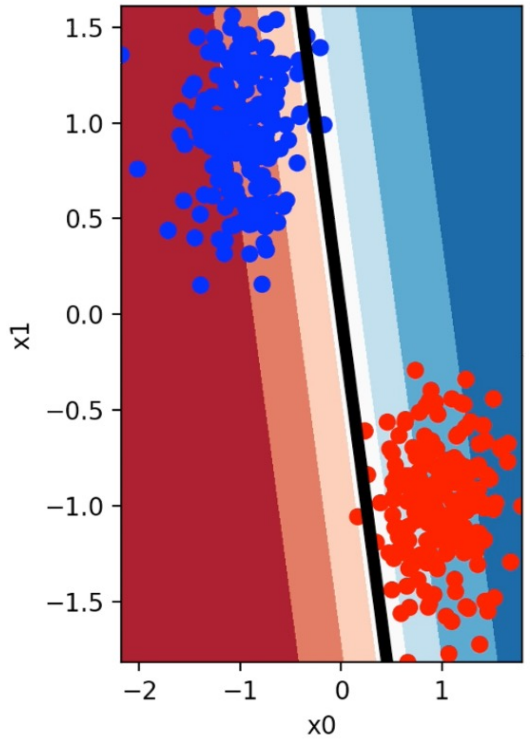
Logistic Regression: Decision Boundary

- We can plot the decision boundary between the positive and negative class as when $w_0 x_0 + w_1 x_1$ is 0 $P(y = 1|\boldsymbol{x}) = 0.5$
  - $w_0 x_0 + w_1 x_1 = 0$
  - $x_1 = -w_0 x_0 / w_1$



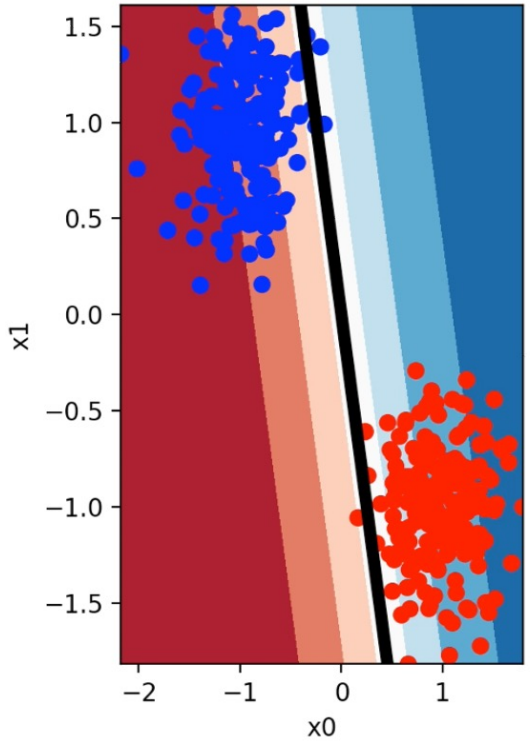
Logistic Regression: Loss Function

- Given the following data and linear model, what is a good loss function to determine how well our model fits the data?



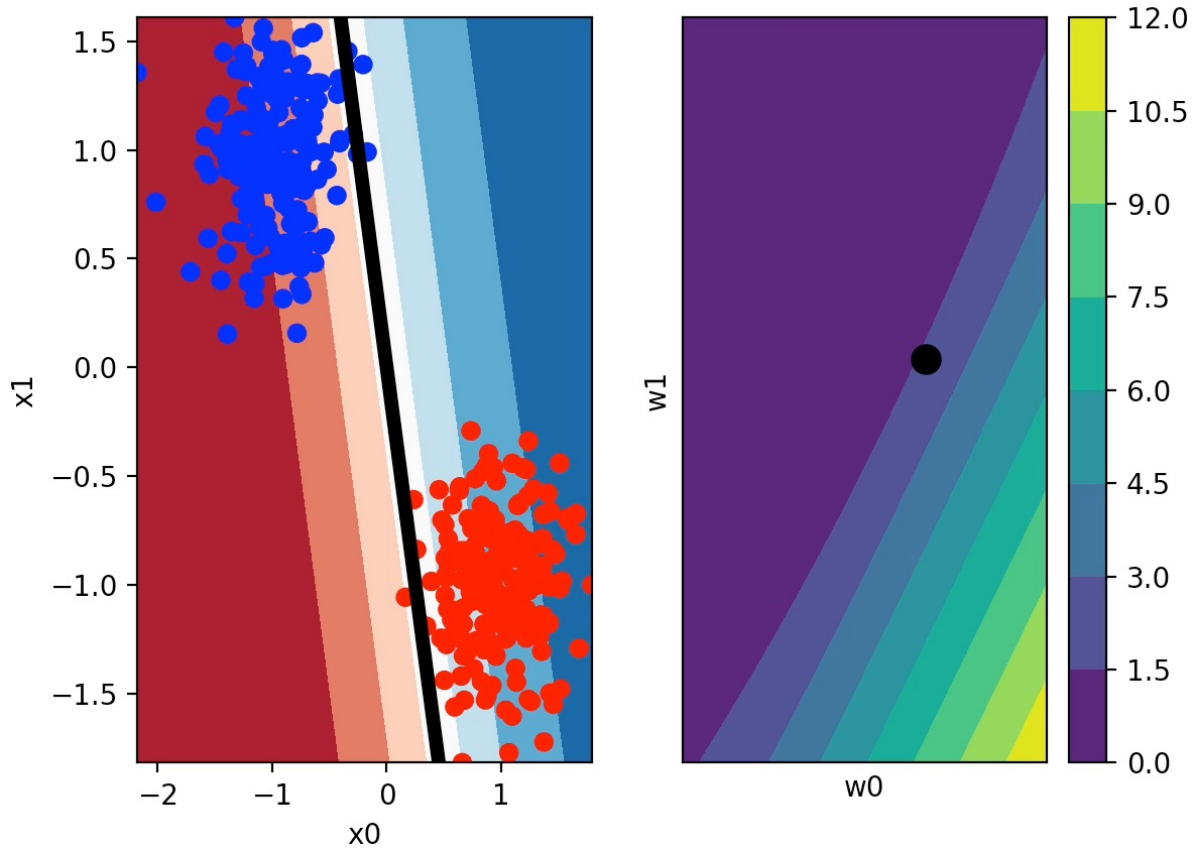
Logistic Regression: Optimization

- We will use what is known as maximum likelihood
- $L(\boldsymbol{w}) = -\left(\sum_{i=1}^{N} \mathrm{y_i} \log \sigma(\boldsymbol{w}^T \boldsymbol{x}_i) + (1 - \mathrm{y_i}) \log(1 - \sigma(\boldsymbol{w}^T \boldsymbol{x}_i))\right)$
  - $\mathrm{y_i}$ can be either 0 and 1
- We can use the same gradient descent algorithm to find a good set of parameters



Logistic Regression: Gradient Descent

- The input is two dimensional
- $P(y = 1|\boldsymbol{x}) = \frac{1}{1+e^{-(w_0 x_0 + w_1 x_1)}}$
- No bias b
- We can plot the decision boundary between the positive and negative class as when $w_0 x_0 + w_1 x_1$ is 0
  $P(y = 1|\boldsymbol{x}) = 0.5$
  - $w_0 x_0 + w_1 x_1 = 0$
  - $x_1 = -w_0 x_0 / w_1$



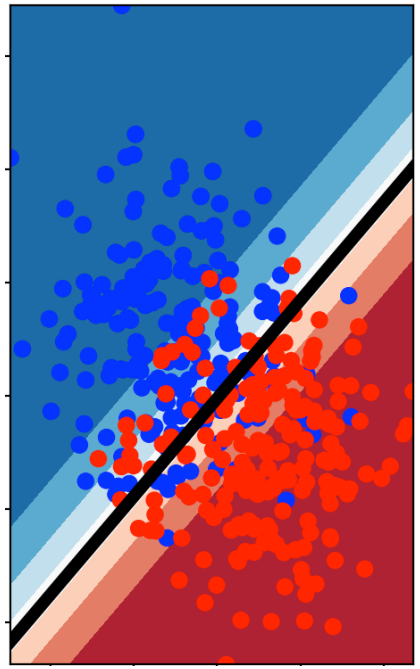
Classes Cannot Always be Perfectly Separated

- In many real-world applications, the classes are not perfectly separated
  - Data could be inherently noisy
  - The predictors may not be informative enough
  - The machine learning model may not be expressive enough
  - The training algorithm used may not be appropriate
- What could happen if your data contains more of one class than another?
  - For example, you want to learn if someone has a rare disease from medical tests
  - Since most people do not have the disease, most examples are of people that do not have the disease

Balanced vs Unbalanced Data

- One should always ensure that they balance their datasets!
  - Every gradient step can sample an equal number of states from each class
  - Or weight the contributions to the loss for each class to account for data being unbalanced
- Is this enough?

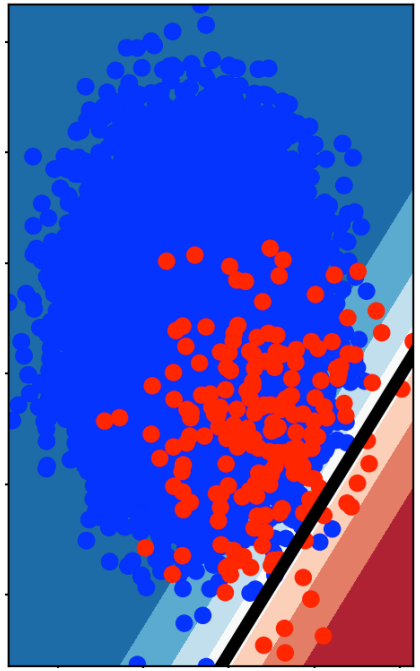
Decision boundary with balanced data



$$P(y = 1|\boldsymbol{x}) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + b)}}$$

Has bias b

Decision boundary with unbalanced data

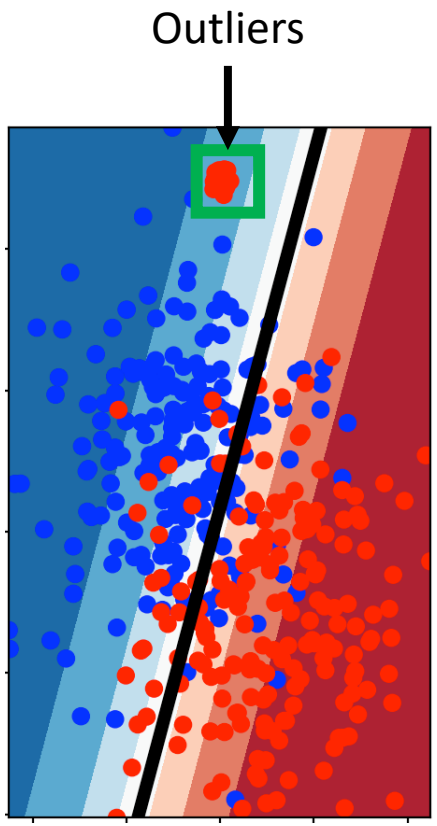


Inductive Bias

- We can significantly reduce the search space by encoding knowledge about the form we think a hypothesis should or should not take
- This can greatly speed up search and result in hypotheses that are easier for humans to understand
- However, if we are not careful, we can accidentally remove valid hypotheses from consideration and, perhaps, this may result in us not finding a solution at all

Balanced vs Unbalanced Data

- Even if the classes themselves are balanced, there may be outliers within those classes
- If these are not explicitly accounted for, the model may ignore them entirely
- For example, a rare disease that affects older people much more than children



Outline

- Supervised Learning
- Linear regression
- Logistic regression
- Explainability
- Hypothesis space

The MNIST Dataset

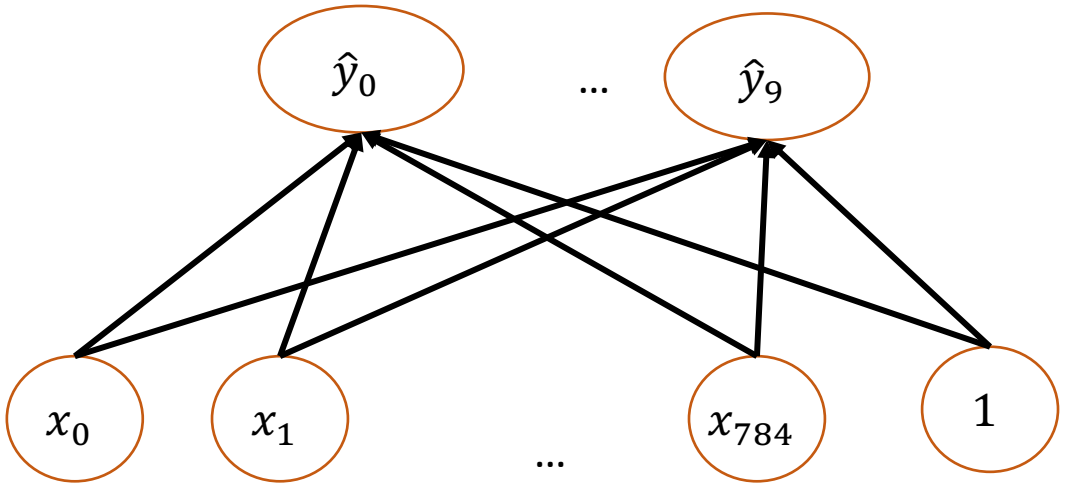
- A dataset of handwritten 28x28 images of digits between 0 and 9
- The objective is to correctly classify each digit
- The dataset consists of 60,000 training images and 10,000 validation images

Softmax Regression

- If we have more than two classes, we can generalize logistic regression
- We got the logistic function from this equation $\frac{e^{\boldsymbol{w}_1^T \boldsymbol{x}}}{e^{\boldsymbol{w}_1^T \boldsymbol{x}} + e^{\boldsymbol{w}_0^T \boldsymbol{x}}}$
- If we have C classes, the probability of class i is $\frac{e^{\boldsymbol{w}_i^T \boldsymbol{x}}}{\sum_{j=1}^{C} e^{\boldsymbol{w}_j^T \boldsymbol{x}}}$

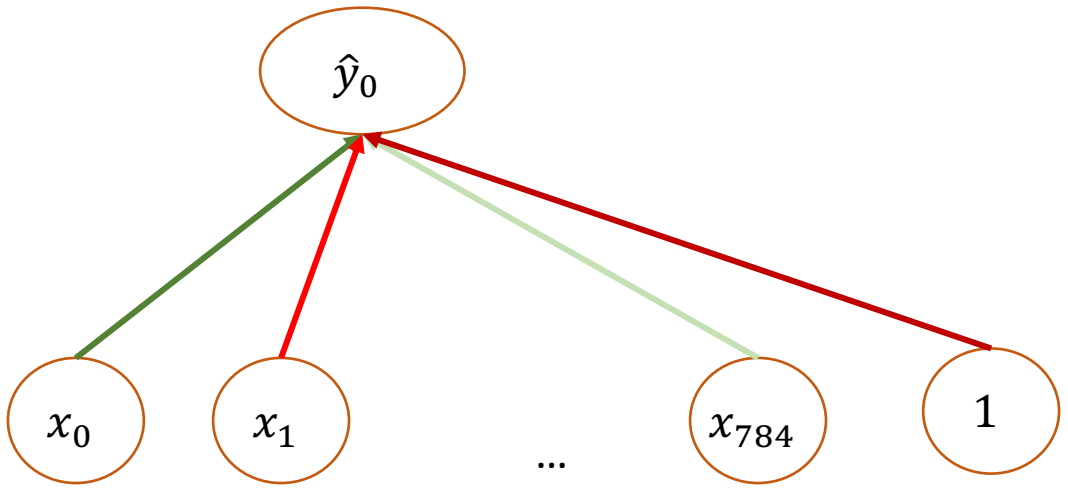
MNIST: Classification

- 784 input features (28x28 images)
- 10 outputs (digits 0-9)
- How might we explain the output of a linear model?

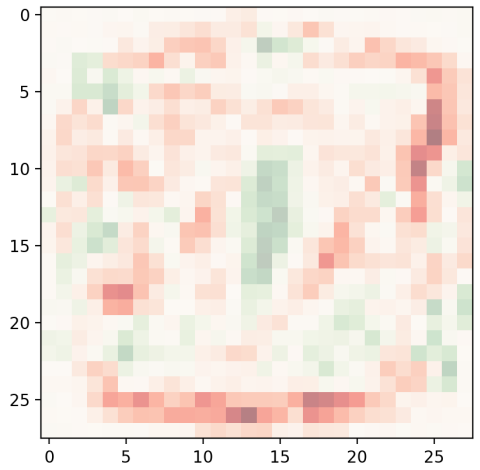


Explanation for a Single Class

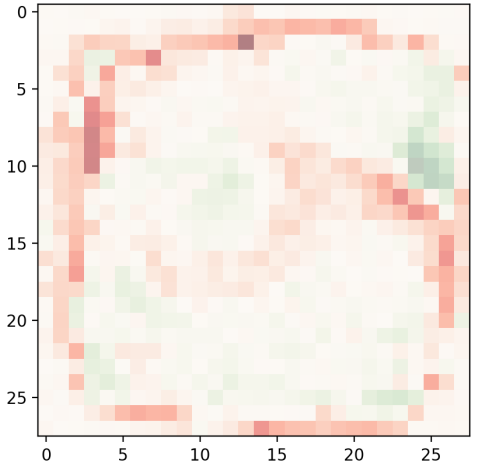
- One can visualize the relative value of the weights
  - Positive weights increase value for class
  - Negative weights decrease value for class
  - However, this does not necessarily increase/decrease the probability since other classes may have weights of higher magnitude for these input features



One



Five

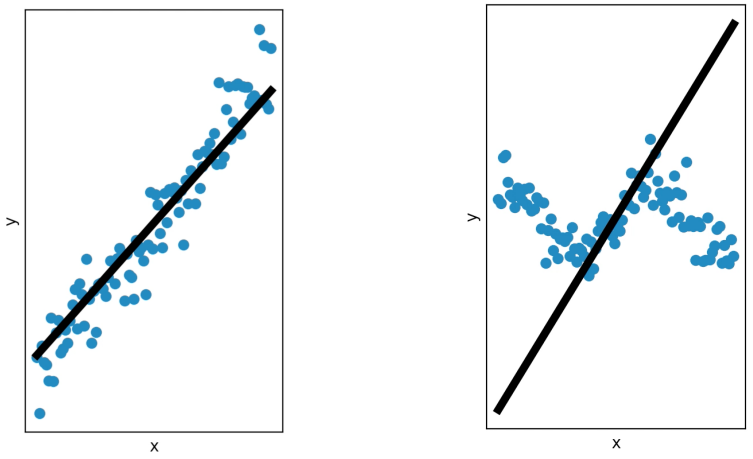


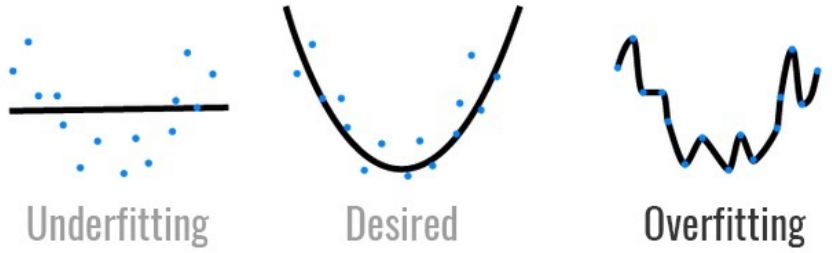
Outline

- Supervised Learning
- Linear regression
- Logistic regression
- Explainability
- Hypothesis space

Hypothesis Space

- The space of all possible hypotheses
  - E.g. The hypothesis space for linear models is all possible lines
- It is important that the hypothesis space be appropriate for the task at hand
  - E.g, if the observations are have a linear input/output relationship, it is best to use a linear model
- However, if the observations have a non-linear input/output relationship, then a linear model will provide a poor explanation of the data
- On the other hand, if your hypothesis space is too large, then you may learn unnecessarily complicated hypotheses





Linear Models: Limitations

- Many interesting problems have a non-linear relationship between the inputs and outputs
- Linear models cannot handle these cases