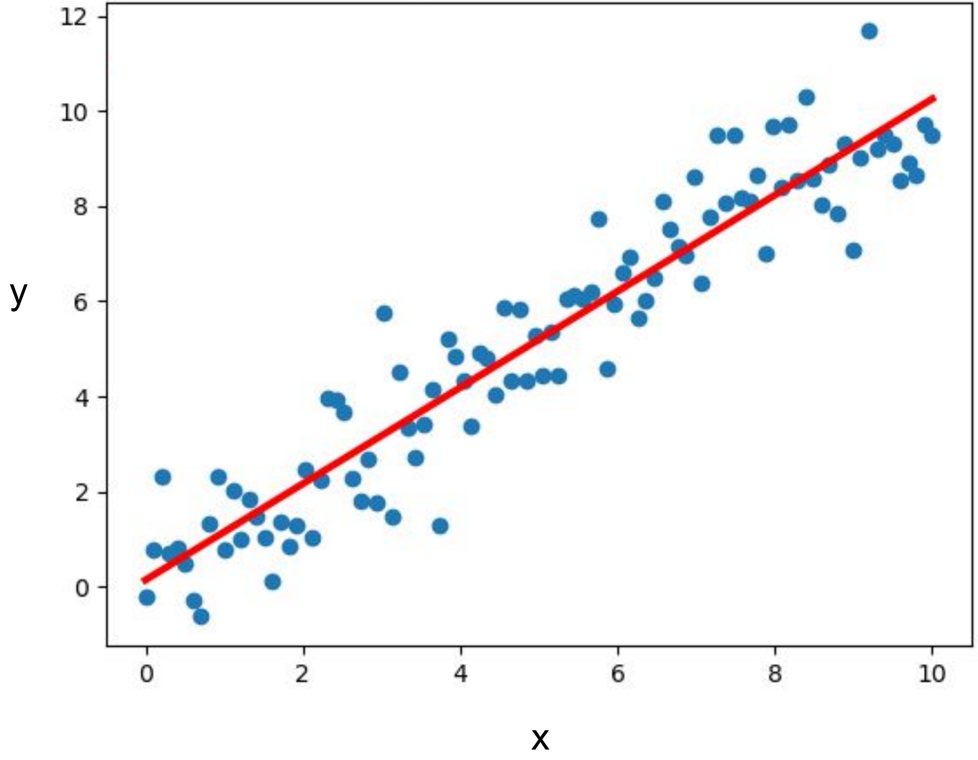
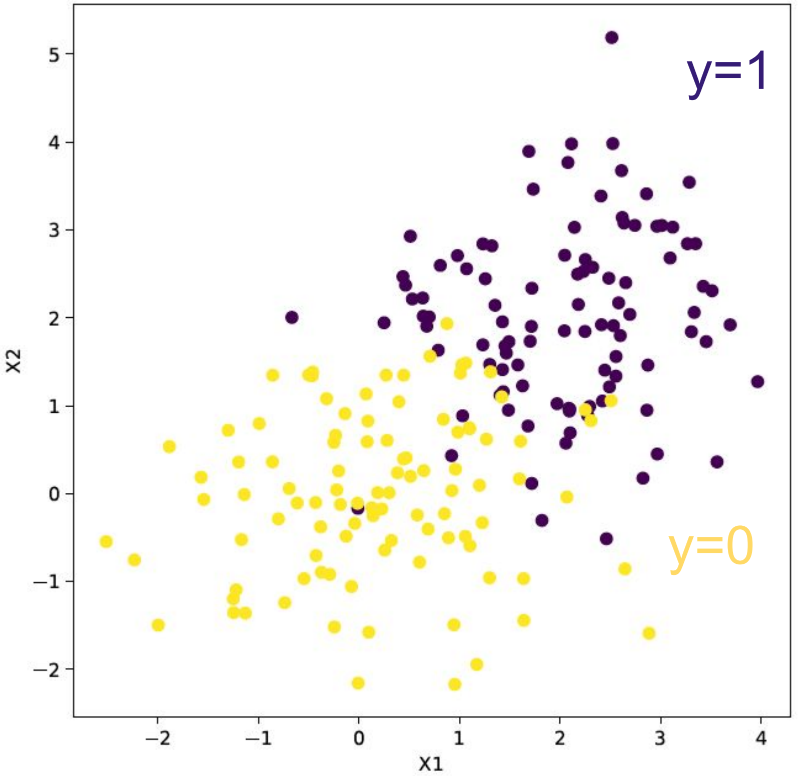
Classification and overfitting

Last time:

regression vs. classification





Last time:

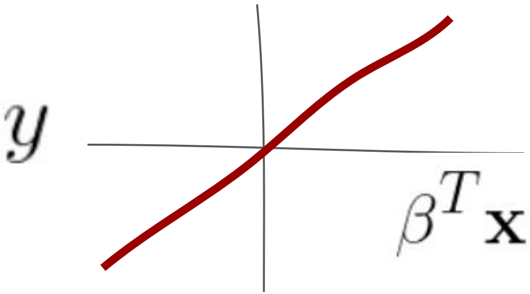
$\mathbf{x}$: vector of k features/covariates/predictors
β : vector of k weights
y : scalar, output/dependent/predicted variable

linear regression: $y = \beta^T \mathbf{x}$

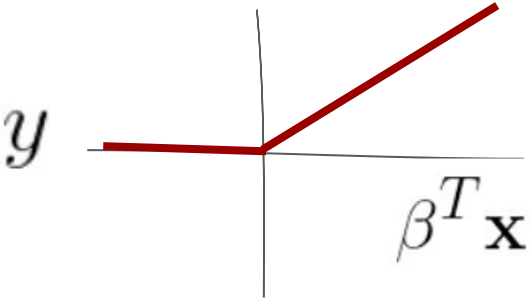
logistic regression: $y = g(\beta^T \mathbf{x})$

Linear and logistic regression are the building blocks for many advanced algorithms, including deep neural networks.

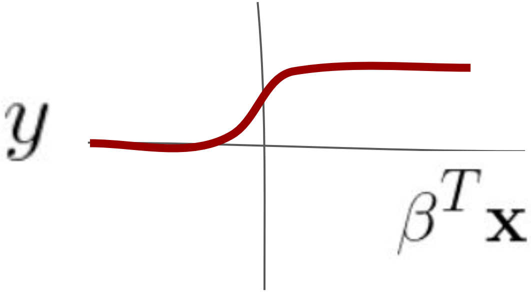
linear function



relu function

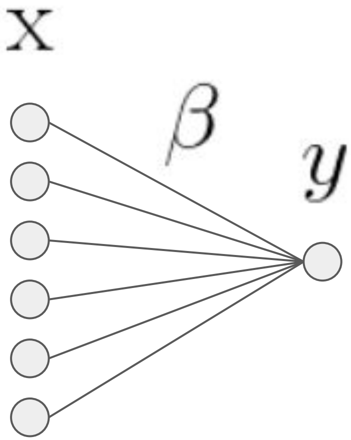


logit/sigmoid function

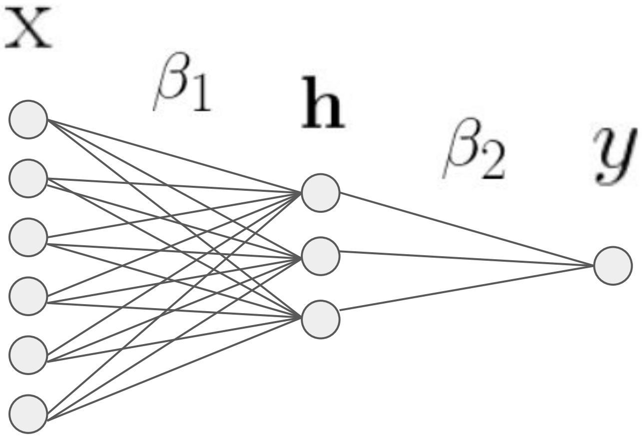


Linear and logistic regression are the building blocks for many advanced algorithms, including deep neural networks.

logistic regression:



deep neural network:



Today

- more practice with logistic regression
- training and testing logistic regression

Try section 1 in Colab notebook.

Training and testing in logistic regression.

We want to know how accurate we can predict y.

Training and testing in logistic regression.

We want to know how accurate we can predict y.

Idea:

1. Take data (X, y).
2. Fit Beta.
3. Report accuracy of how well y_hat predicts y.

Training and testing in logistic regression.

We want to know how accurate we can predict y.

Idea:

1. Take data (X, y).
2. Fit Beta.
3. Report accuracy of how well y_hat predicts y.

Can we trust this reported accuracy?

Training and testing in logistic regression.

When you have many features (e.g., 100k fMRI voxels) and few samples (e.g., 100 trials), machine learning algorithms will overfit.

Training and testing in logistic regression.

When you have many features (e.g., 100k fMRI voxels) and few samples (e.g., 100 trials), machine learning algorithms will overfit.

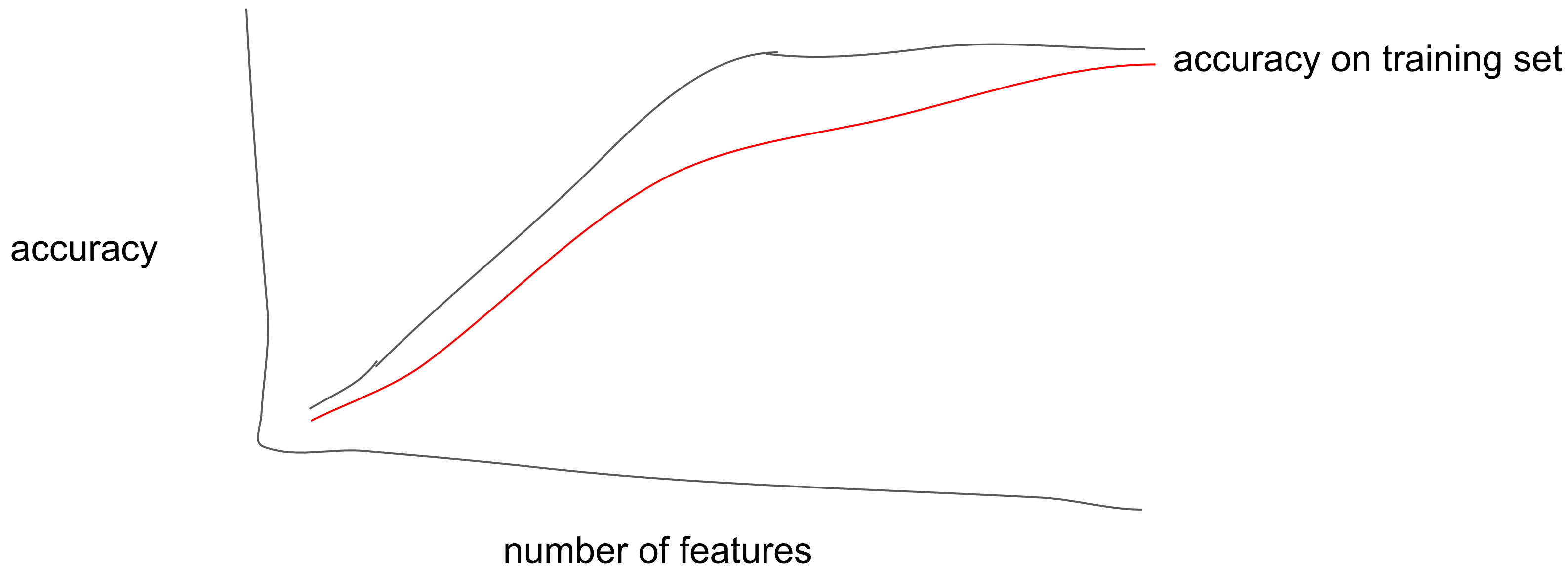
→ They find spurious correlations in the training data that do *not* generalize to new data.

Training and testing in logistic regression.

When you have many features (e.g., 100k fMRI voxels) and few samples (e.g., 100 trials), machine learning algorithms will overfit.

→ They find spurious correlations in the training data that do *not* generalize to new data.

Let’s test this out in the Colab Notebook.


accuracy on training set
accuracy
number of features

Preventing overfitting is a main focus in machine learning.

However, before we can prevent overfitting, we need to make sure our reported accuracies are not inflated from overfitting.

Preventing overfitting is a main focus in machine learning.

However, before we can prevent overfitting, we need to make sure our reported accuracies are not inflated from overfitting.

Idea: Separate data into two subsets:
a training set (fit our weights with this)
and a test set (compute accuracy with this set)

Preventing overfitting is a main focus in machine learning.

However, before we can prevent overfitting, we need to make sure our reported accuracies are not inflated from overfitting.

Idea: Separate data into two subsets:
a training set (fit our weights with this)
and a test set (compute accuracy with this set)

Basic advice:
Given a dataset, split it immediately. Then never touch the test set until you are ready to report accuracy.

Let’s see what happens when we report accuracy on a heldout set (not the training set) in the Colab notebook.