CSC411 – Midterm Review

Yingzhou(Joe) Wu

University of Toronto

2018 Winter

ML basic terms (relevant to midterm)

Regression

Classification

Binary classification

Multi-class classification

Overfitting

Underfitting

Generalization

Regularization

Maximum likelihood

Maximize a posteriori

Stochastic gradient descent

Steepest gradient descent

Training Data

Validation Data

Test Data

Discriminative approach

Generative approach

Bayesian approach

Sample question 1

Given a discriminative model with parameters θ and training data pairs x; y.

- The likelihood is ___________________
- MAP estimation maximizes _________ x _________

Sample question 2

Suppose you have a dataset of labeled examples for training a machine learning system.
(A). (4 points) Define a validation set.
(B). (4 points) Describe the trade-offs involved in assigning examples to the validation set versus the training set

Linear Regression

Inputs (features) and output

Inputs : Vector $X \in \mathbb{R}^d$

Output : $y \in \mathbb{R}$

Model (Parameters)

$$\hat{y} = w_0 + w_1 x_1 + \ldots w_d x_d \text{ for } \mathbf{w} \in \mathbb{R}^{d+1}$$

Loss function

$$L_2(y, \hat{y}) = (y - \hat{y})^2$$

$$L_1(y, \hat{y}) = |y - \hat{y}|$$

Regularization

$$L_2 : \frac{\lambda}{2} \sum_{i=0}^{d} w_i^2$$

$$L_1 : \lambda \sum_{i=0}^{d} |w_i|$$

Linear Regression (Sample Questions)

Sample question 1

True or False ?

Assume that you have training data with continuous features and targets. Linear regression trained with L2 loss is robust to outliers in the training data.

Sample question 2

Write down the L1 and L2 loss function and L1 and L2 regularization term, if assume the residue follows Gaussian distribution, then maximum likelihood is equivalent to L1 or L2 loss function?

Linear Classification and Logistic Regression

Inputs (features) and output

Inputs : Vector $X \in \mathbb{R}^d$ Output : $y \in \{0,1\}$

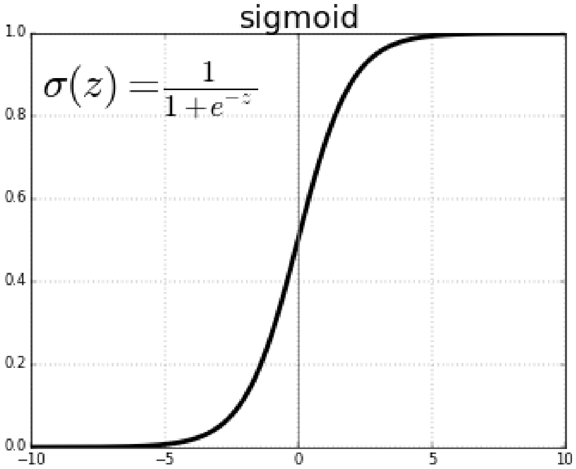
Model (Parameters)

Linear Classification

$\hat{y} = w_0 + w_1 x_1 + \ldots w_d x_d$ for $\mathbf{w} \in \mathbb{R}^{d+1}$

Logistic Regression

$\hat{y} = \sigma(w_0 + w_1 x_1 + \ldots w_d x_d)$ for $\mathbf{w} \in \mathbb{R}^{d+1}$, where $\sigma(t) = 1/(1 + e^{-t})$

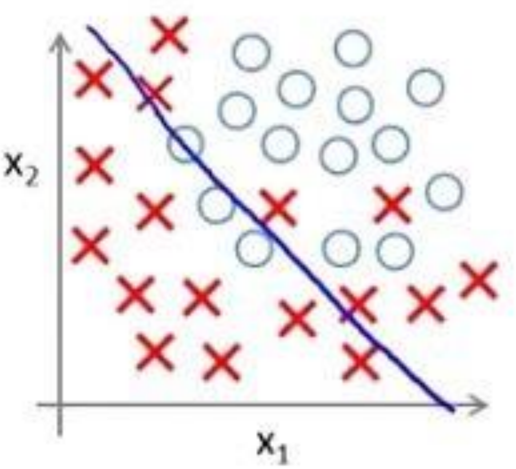


Discriminative approach

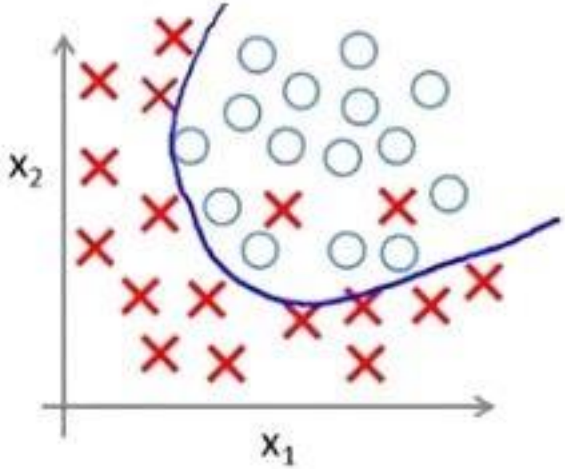
Logistic Regression (Sample Questions)

Sample question 1

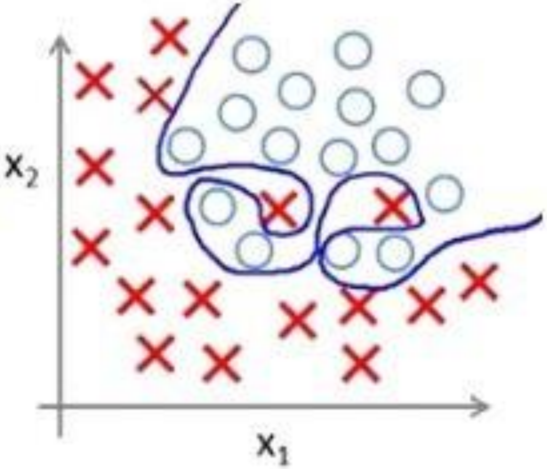
Which is the decision boundary for logistic regression?



A



B



C

Sample question 2

True or False?

1. Logistic Regression use Maximum likelihood and gradient descent to learn weights.

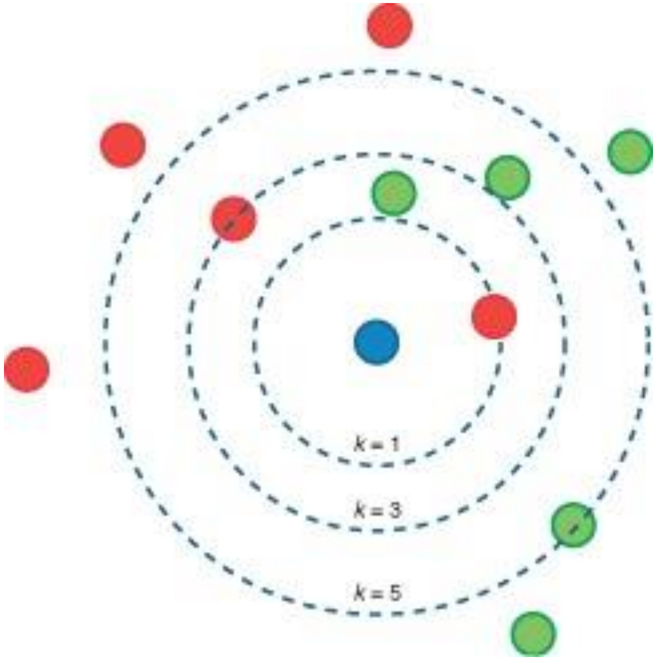
2. Logistic Regression only can be used for binary classification.

3. Logistic Regression learn the joint probability distribution of features and the dependent variable.

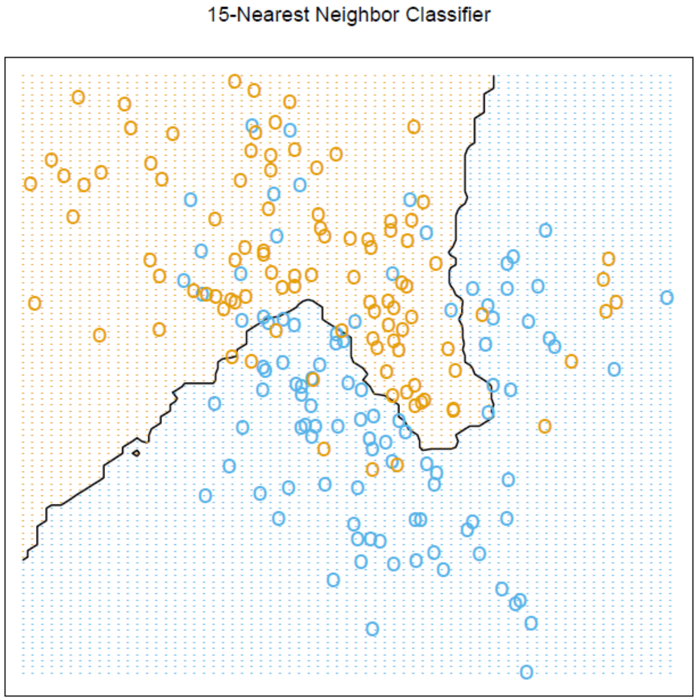
KNN (K nearest neighbors)

Nonparametric

Nonlinear decision boundary



k is a hyperparameter



KNN (Sample Questions)

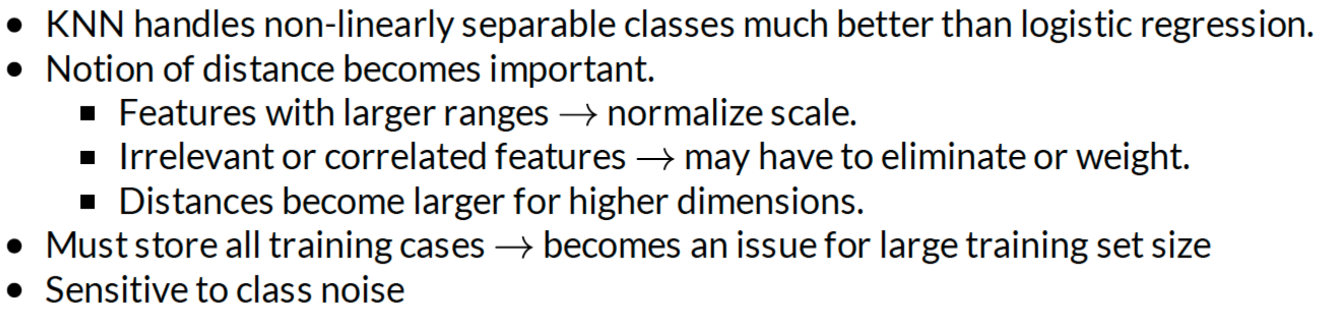
Sample question 1

Assume we are preprocessing our data using an invertible linear transformation on the features of our training data. The transformation can either be some orthogonal (i.e. rotations) matrix or some diagonal matrix. Say if this can have any effect on the performance of the following algorithms, and explain in no more than two sentences.

- Orthogonal preprocessing on nearest neighbor classification.
- Diagonal preprocessing on nearest neighbor classification.

Sample question 2

What kinds of data are expected to be (in)appropriate for a k nearest neighbor classifier?



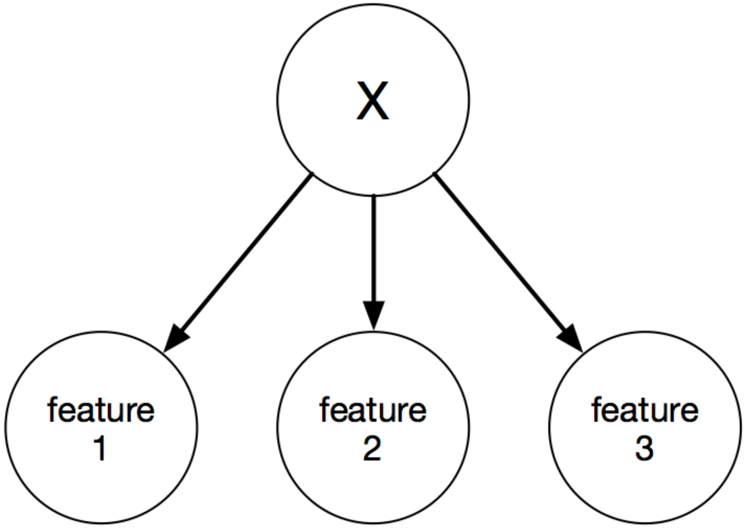
- KNN handles non-linearly separable classes much better than logistic regression.
- Notion of distance becomes important.
  - Features with larger ranges → normalize scale.
  - Irrelevant or correlated features → may have to eliminate or weight.
  - Distances become larger for higher dimensions.
- Must store all training cases → becomes an issue for large training set size
- Sensitive to class noise

Naïve Bayes

Generative approach

What is Naïve?

$$\overset{\text{Posterior}}{P(c|x)} = \frac{\overset{\text{Likelihood}}{P(x|c)}\overset{\text{Class prior}}{P(c)}}{P(x)}$$



$$P(c|X) = P(x_1|x)P(x_2|c) \ldots P(x_n|c)P(c)/P(x)$$

Naïve Bayes(Sample Questions)

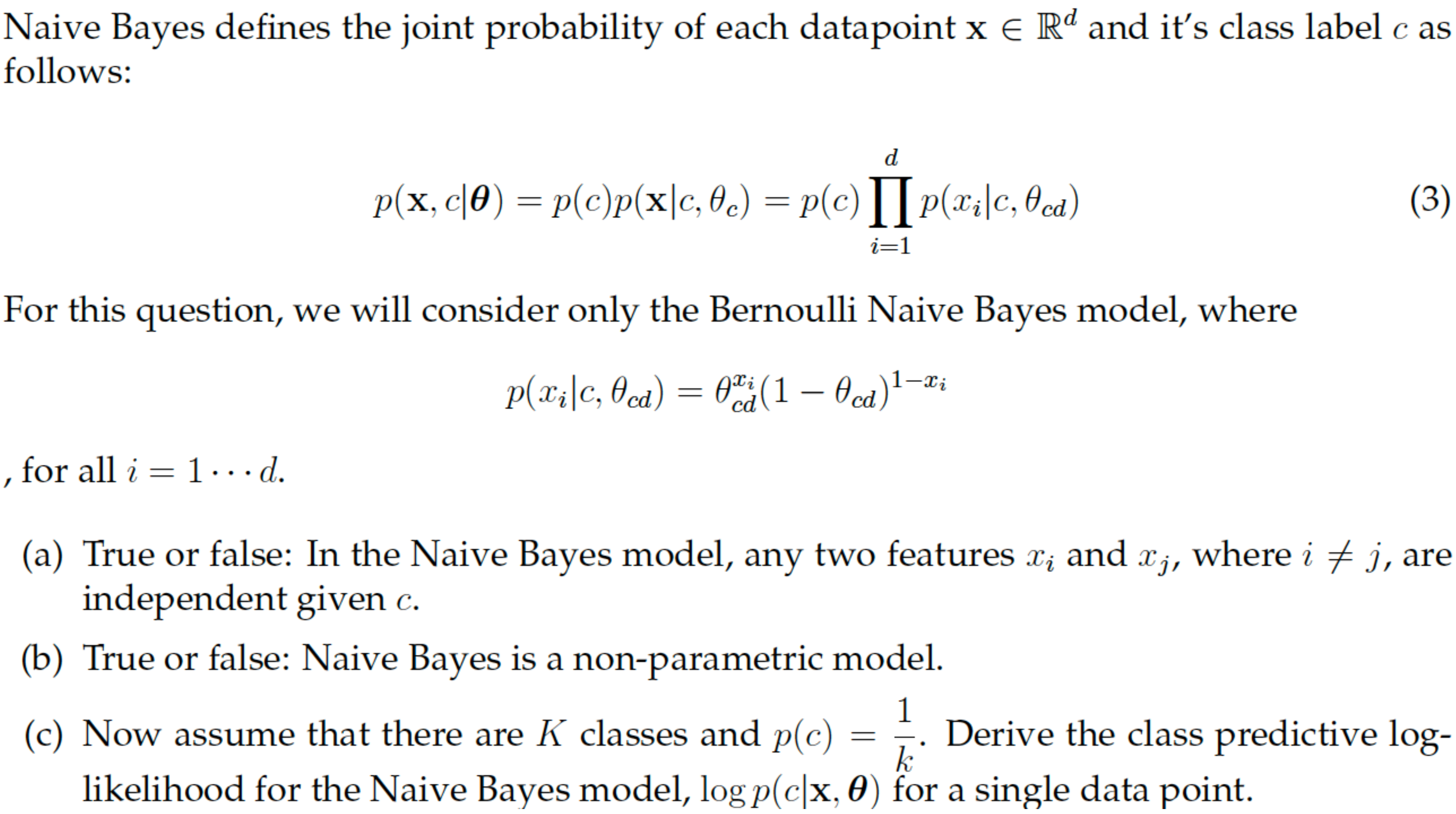
Sample question 1

Imagine that you want to decide if the Leafs are going to win or lose their next hockey game. You are going to base this decision on a dataset that contains the following data for each of their last 1000 games: opponent, day of the week, location, and outcome. You have the same information for the next game, except the outcome.

(B). (6 points)Write down the decision rule equation for the a Naive Bayes classifier for this problem. Define the key simplifying assumption in the Naive Bayes method, and explain why you think it is or is not applicable here.

Naïve Bayes(Sample Questions)

Sample question 2



Naive Bayes defines the joint probability of each datapoint $\mathbf{x} \in \mathbb{R}^d$ and it's class label c as follows:

$$p(\mathbf{x}, c|\boldsymbol{\theta}) = p(c)p(\mathbf{x}|c, \theta_c) = p(c)\prod_{i=1}^{d} p(x_i|c, \theta_{cd}) \tag{3}$$

For this question, we will consider only the Bernoulli Naive Bayes model, where

$$p(x_i|c, \theta_{cd}) = \theta_{cd}^{x_i}(1 - \theta_{cd})^{1-x_i}$$

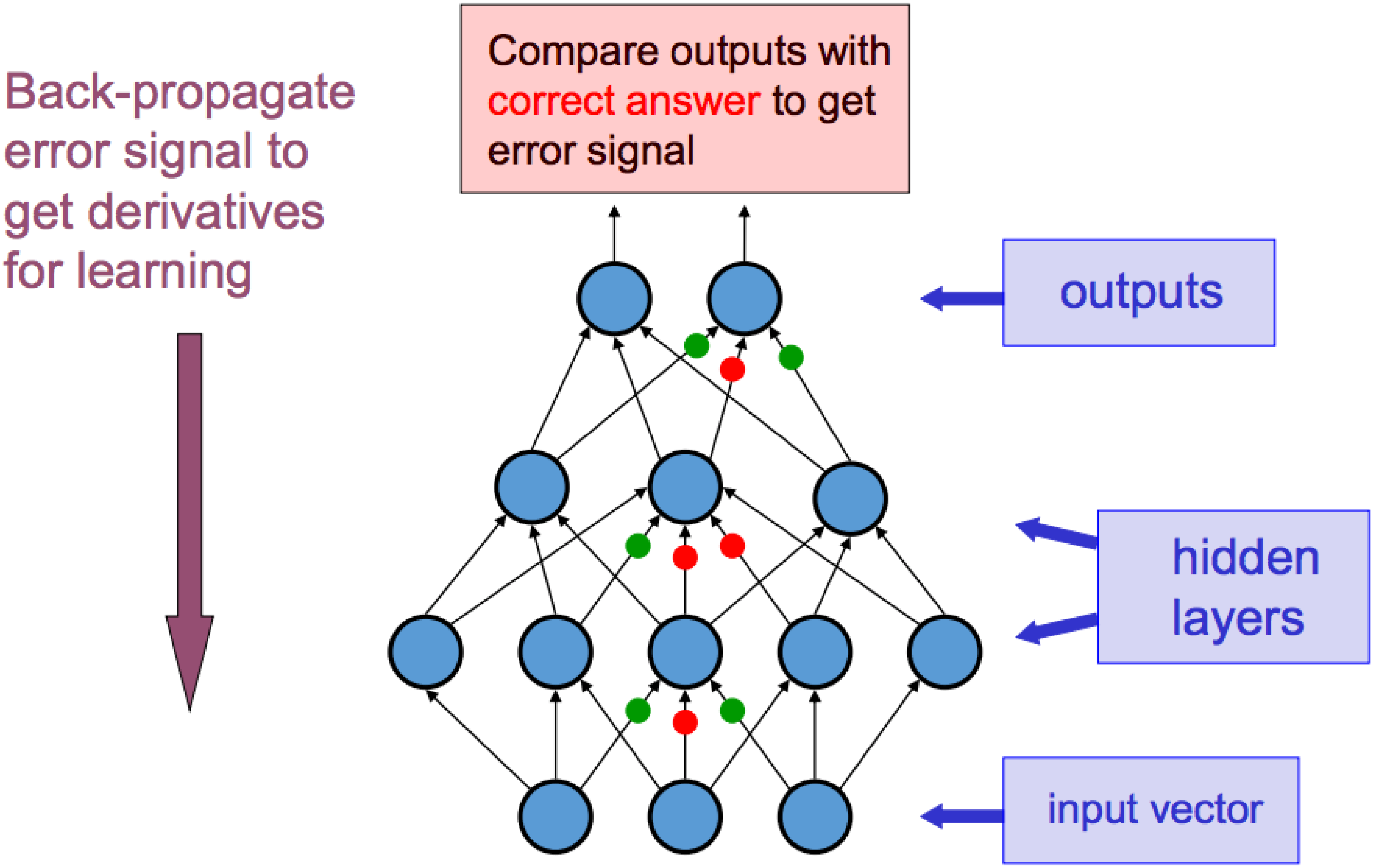
, for all $i = 1 \cdots d$.

(a) True or false: In the Naive Bayes model, any two features x_i and x_j, where $i \neq j$, are independent given c.

(b) True or false: Naive Bayes is a non-parametric model.

(c) Now assume that there are K classes and $p(c) = \frac{1}{k}$. Derive the class predictive log-likelihood for the Naive Bayes model, $\log p(c|\mathbf{x}, \boldsymbol{\theta})$ for a single data point.

Neuron Network



Back-propagate error signal to get derivatives for learning

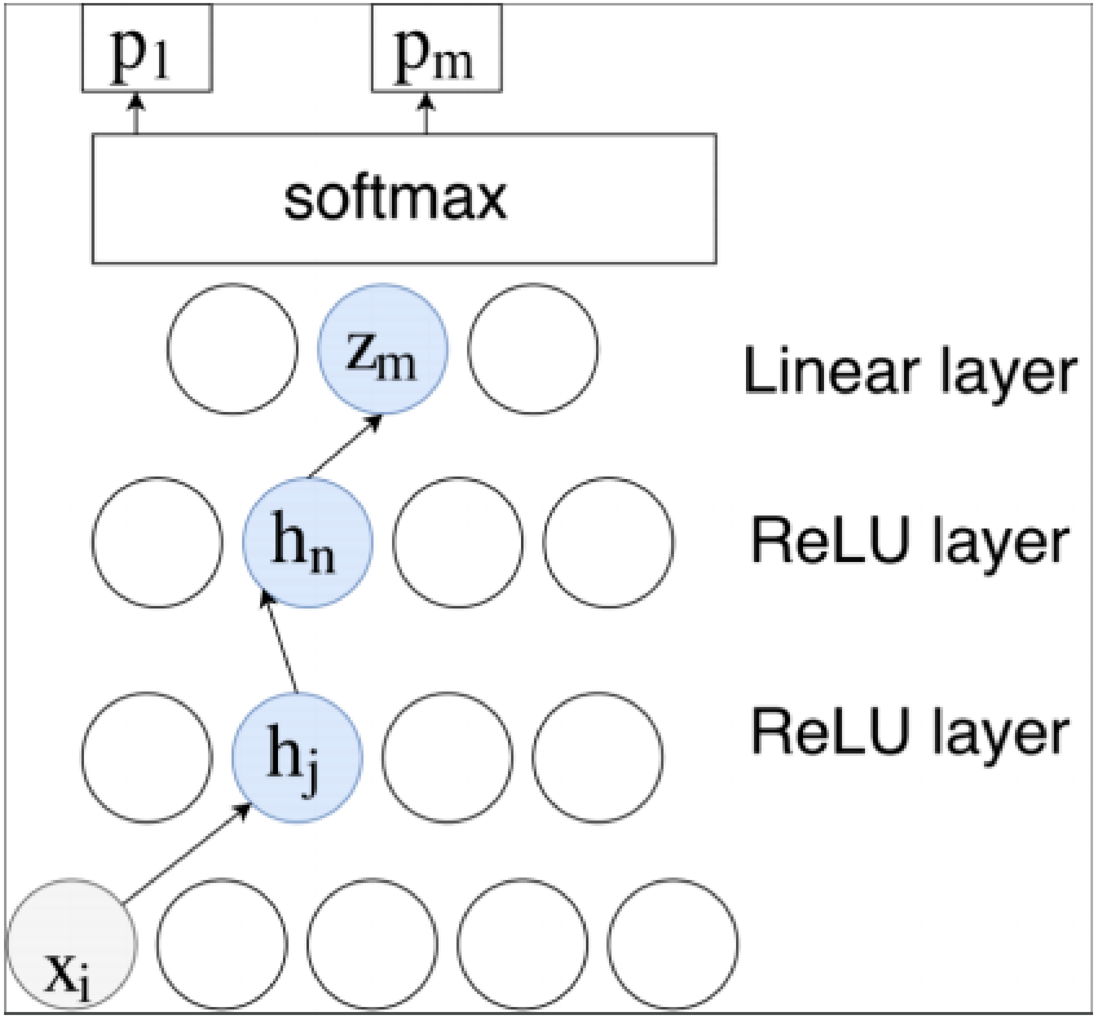
Compare outputs with correct answer to get error signal

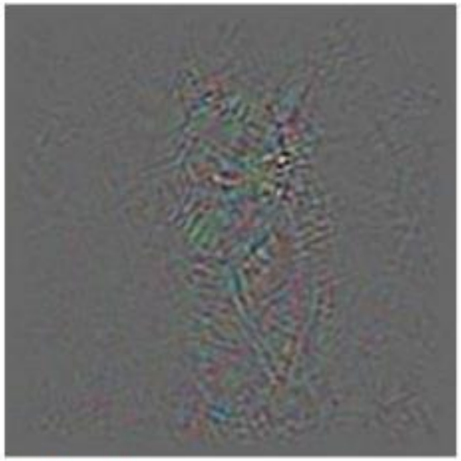
outputs

hidden layers

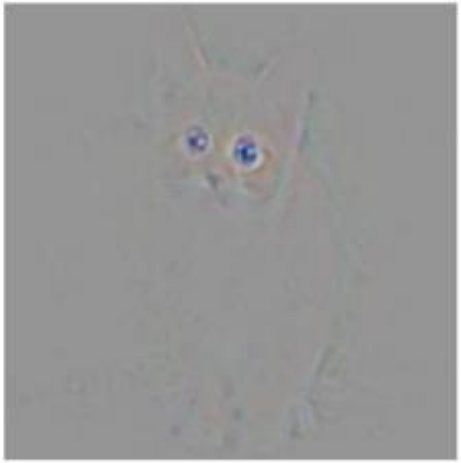
input vector

Guided backpropagation





Backprop



Guided Backprop

Neuron Network (Sample Questions)

Sample question 1

True or False?

1) A perceptron is guaranteed to perfectly learn a given linearly separable function within a finite number of training steps.

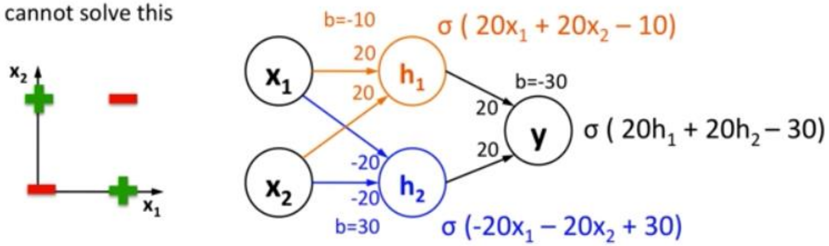
2) For effective training of a neural network, the network should have at least 5-10 times as many weights as there are training samples.

3) A single perceptron can compute the XOR function.

Sample question 2

Draw a neural network to compute **XOR**

Linear classifiers cannot solve this



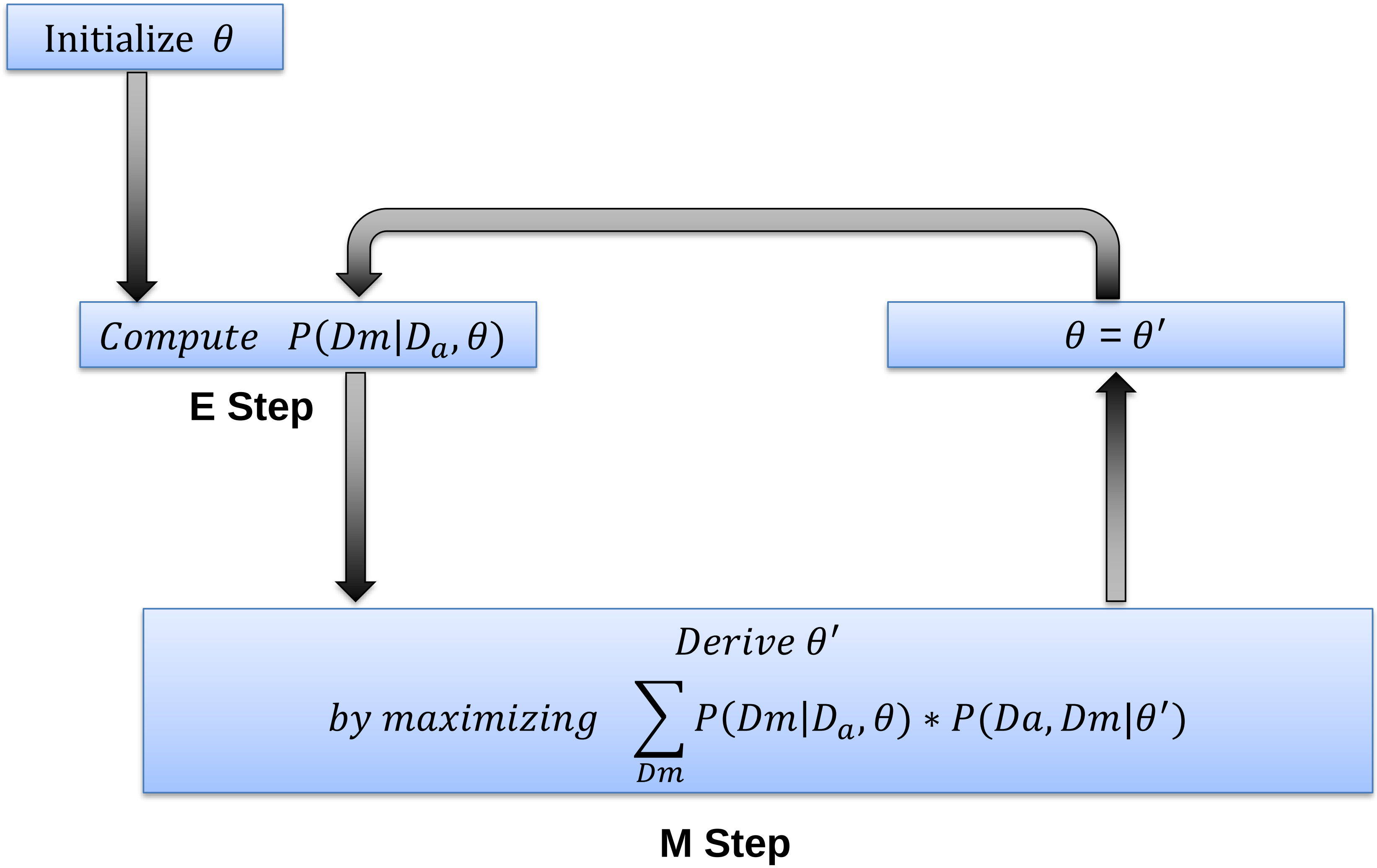
x_2
x_1
+
−
−
+
b=-10
$\sigma(20x_1 + 20x_2 - 10)$
20
x_1
20
h_1
b=-30
20
y
$\sigma(20h_1 + 20h_2 - 30)$
20
-20
x_2
-20
h_2
b=30
$\sigma(-20x_1 - 20x_2 + 30)$

$\sigma(20*0 + 20*0 - 10) \approx 0$	$\sigma(-20*0 - 20*0 + 30) \approx 1$	$\sigma(20*0 + 20*1 - 30) \approx 0$
$\sigma(20*1 + 20*1 - 10) \approx 1$	$\sigma(-20*1 - 20*1 + 30) \approx 0$	$\sigma(20*1 + 20*0 - 30) \approx 0$
$\sigma(20*0 + 20*1 - 10) \approx 1$	$\sigma(-20*0 - 20*1 + 30) \approx 1$	$\sigma(20*1 + 20*1 - 30) \approx 1$
$\sigma(20*1 + 20*0 - 10) \approx 1$	$\sigma(-20*1 - 20*0 + 30) \approx 1$	$\sigma(20*1 + 20*1 - 30) \approx 1$

EM Algorithms

Available DATA (D_a), Missing DATA (D_m), D = D_a + D_m

MODEL (θ)



Initialize θ

$Compute \quad P(Dm|D_a, \theta)$

E Step

$\theta = \theta'$

$Derive\ \theta'$

$by\ maximizing \quad \sum_{Dm} P(Dm|D_a, \theta) * P(Da, Dm|\theta')$

M Step

Generalization and Regularization

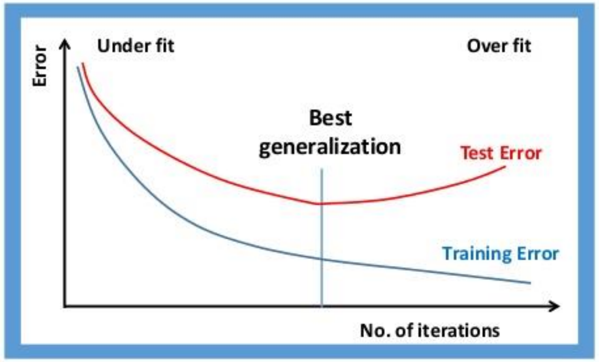
Generalization

In supervised learning applications in machine learning and statistical learning theory, **generalization error** (also known as the **out-of-sample error**) is a measure of how accurately an algorithm is able to predict outcome values for previously unseen data

Regularization

In mathematics and statistics, particularly in the fields of machine learning and inverse problems, regularization is a process of introducing additional information in order to solve an ill-posed problem or to prevent overfitting.

Generalization



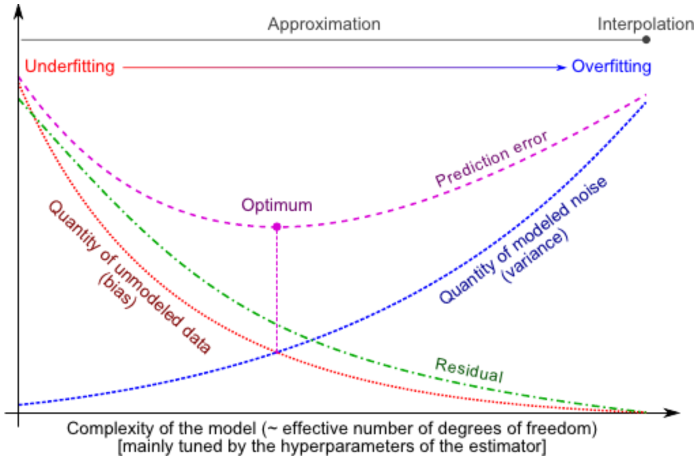
Overfitting and Underfitting

Overfitting

a statistical model describes random error or noise instead of the underlying relationship. Overfitting occurs when a model is excessively complex, such as having too many parameters relative to the number of observations. A model that has been overfitted has poor predictive performance, as it overreacts to minor fluctuations in the training data.

Underfitting

Underfitting occurs when a statistical model or machine learning algorithm cannot capture the underlying trend of the data. Underfitting would occur, for example, when fitting a linear model to non-linear data. Such a model would have poor predictive performance.



Maximum likelihood and MAP

Maximum Likelihood

maximum likelihood estimation (**MLE**) is a method of estimating the parameters of a statistical model given observations, by finding the parameter values that maximize the likelihood of making the observations given the parameters. MLE can be seen as a special case of the maximum a posteriori estimation(MAP) that assumes a uniform prior distribution of the parameters, or as a variant of the MAP that ignores the prior and which therefore is unregularized.

$$\hat{\theta}_{\mathrm{ML}}(x) = \underset{\theta}{\arg\max}\, f(x \mid \theta)$$

Maximize A Posteriori

MAP is closely related to the method of maximum likelihood (ML) estimation, but employs an augmented optimization objective which incorporates **a prior distribution (that quantifies the additional information available through prior knowledge of a related event)** over the quantity one wants to estimate. MAP estimation can therefore be seen as a regularization of ML estimation.

$$\hat{\theta}_{\mathrm{MAP}}(x) = \underset{\theta}{\arg\max}\, f(\theta \mid x) = \underset{\theta}{\arg\max}\, \frac{f(x \mid \theta)\, g(\theta)}{\displaystyle\int_{\vartheta} f(x \mid \vartheta)\, g(\vartheta)\, d\vartheta} = \underset{\theta}{\arg\max}\, f(x \mid \theta)\, g(\theta).$$

Steepest and Stochastic Gradient Descent

Steepest Gradient Descent

Gradient descent is based on the observation that if the multi-variable function Q(w) is differentiable in a neighborhood of a point a, then Q(w) decreases fastest if one goes from a in the direction of the negative gradient of Q at w.

$$w := w - \eta \nabla Q(w) = w - \eta \sum_{i=1}^{n} \nabla Q_i(w)/n,$$

Stochastic Gradient Descent

Stochastic gradient descent (often shortened to SGD), also known as incremental **gradient descent**, is a **stochastic** approximation of the **gradient descent** optimization and iterative method for minimizing an objective function that is written as a sum of differentiable functions.

For $i = 1, 2, \ldots, n$

$$w := w - \eta \nabla Q_i(w).$$

Data

Training Data

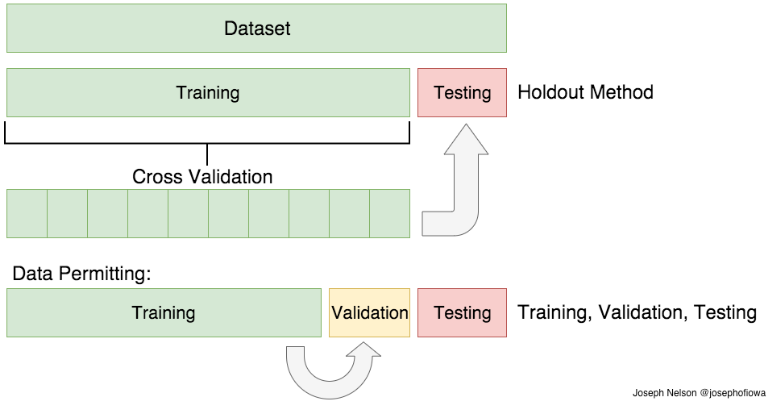
The sample of data used to fit the model.

Validation Data

The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration.

Test Data

The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.



Dataset

Training | Testing | Holdout Method

Cross Validation

Data Permitting:

Training | Validation | Testing | Training, Validation, Testing

Joseph Nelson @josephofiowa

ML Approaches

Discriminative

- Discriminative approach. Fit $P(y|x;\theta)$ by some parametric model.

Generative

- Generative approach. Fit $P(x,y;\theta)$ by some parametric model, and use it to determine $P(y|x;\theta)$.

Bayesian

- Bayesian approach. Instead of a single model θ we have a distribution over θ, $p(\theta)$ so $p(y|x) = \int p(y|x,\theta)p(\theta)$