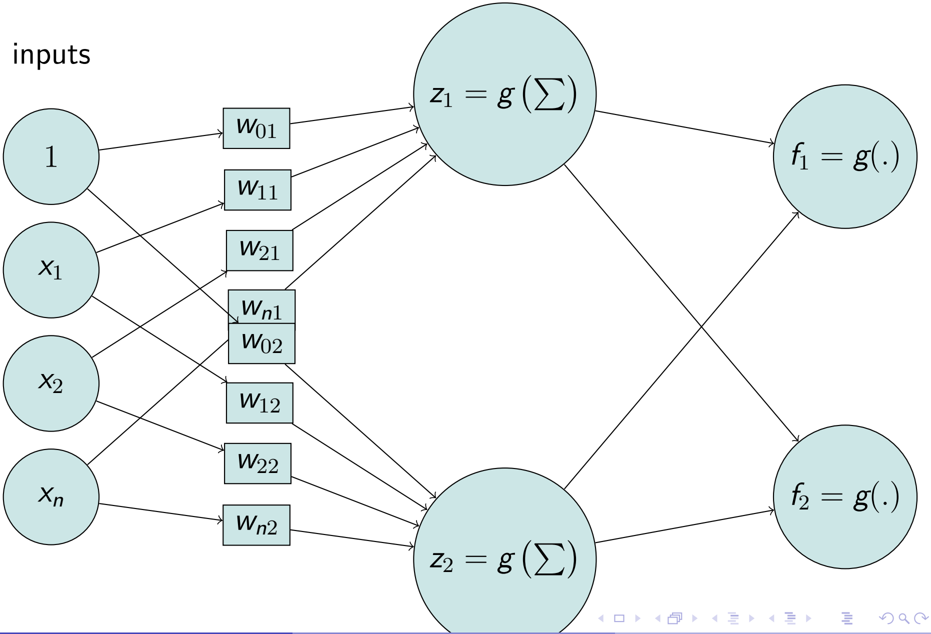
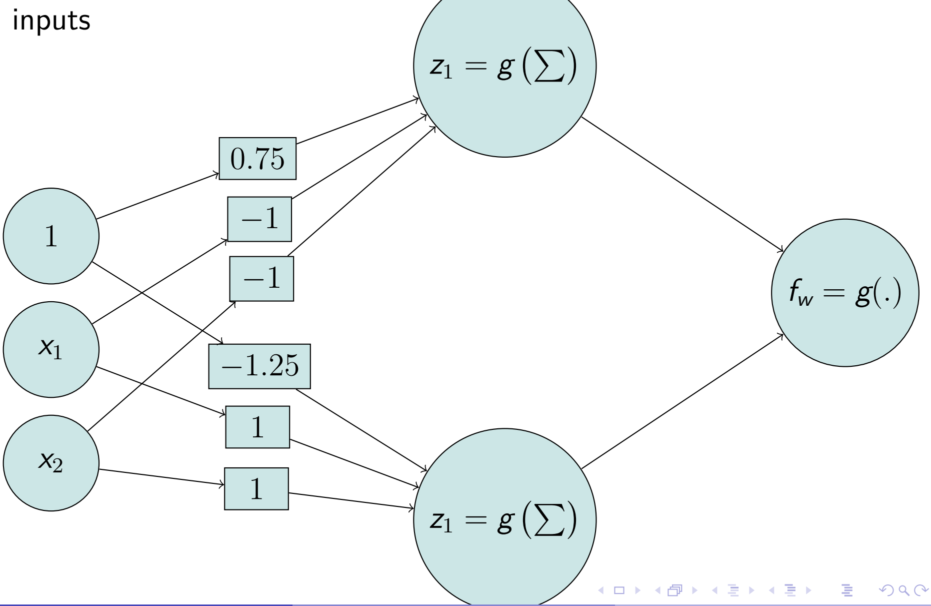
Lecture 17: Training Neural Networks, Logistic Regression

Instructor: Prof. Ganesh Ramakrishnan

Feed-forward Neural Nets



Eg: Feed-forward Neural Net for XOR



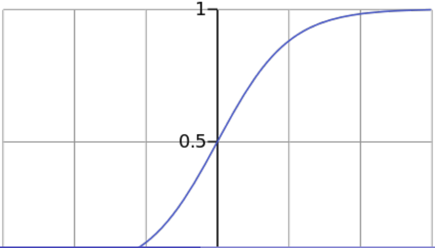
Training a Neural Network

STEP 0: Pick a network architecture

- Number of input units: Dimension of features $\phi\left(x^{(i)}\right)$.
- Number of output units: Number of classes.
- Reasonable default: 1 hidden layer, or if >1 hidden layer, have same number of hidden units in every layer.
- Number of hidden units in each layer a constant factor (3 or 4) of dimension of x.
- We will use
  - the smooth sigmoidal function $g(s) = \frac{1}{1+e^{-s}}$: **How do we train a single node sigmoidal neural network?**
  - instead of the non-smooth step function $g(s) = 1$ if $s \in [\theta, \infty)$ and $g(s) = 0$ otherwise: **Single node step function neural network is perceptron, which we know how to train.**

Training for single node sigmoidal NN

1. Neural Networks: Cascade of layers of sigmoidal perceptrons giving you smoothness and non-linearity
2. Single node sigmoidal NN is also called **(Binary) Logistic Regression**, abbreviated as **LR**
   - $sign\left((w^*)^T\phi(x)\right)$ replaced by $g\left((w^*)^T\phi(x)\right)$ where $g(s)$ is sigmoid function: $g(s) = \frac{1}{1+e^{-s}}$
3. $g\left((w^*)^T\phi(x)\right) = \frac{1}{1+e^{-(w^*)^T\phi(x)}} \in [0,1]$ can be intepreted as $Pr(y=1|x)$
   - Then $Pr(y=0|x) = ?$



Probability theory review in context of LR

- Sample space(S): A sample space is defined as a set of all possible outcomes of an experiment. For LR: $S = \{all\ possible\ examples\ x\ with\ class\ y\}$. $|S| = \infty$
- Event (E) : An event is defined as any subset of the sample space. Total number of distinct events possible is $2^{|S|}$, where $|S|$ is the number of elements in the sample space.
- Random variable: A random variable is a mapping (or function) from set of events to a set of real numbers.
  $\phi(.)$ is a **continuous** random vector

$$\phi(.) : 2^S \rightarrow \mathbb{R}^p$$

  Y is a **discrete** random (class) variable mapping events to a countable set $\{0, 1\}$

$$Y : 2^S \rightarrow \{0, 1\}$$

Axioms of Probability

- For every event E, $0 \leq Pr(E) \leq 1$
- $Pr(S) = 1$
- If $E_1, E_2, \ldots, E_n$ is a set of pairwise disjoint events, then

$$Pr(\bigcup_{i=1}^{n} E_i) = \sum_{i=1}^{n} Pr(E_i)$$

Bayes' Theorem

Let $B_1, B_2, ..., B_n$ be a set of mutually exclusive events that together form the sample space S. Let A be any event from the same sample space, such that $P(A) > 0$. Then,

$$Pr(B_i/A) = \frac{Pr(B_i \cap A)}{Pr(B_1 \cap A) + Pr(B_2 \cap A) + \cdots + Pr(B_n \cap A)} \tag{1}$$

Using the relation $P(B_i \cap A) = P(B_i) \cdot P(A/B_i)$

$$Pr(B_i/A) = \frac{Pr(B_i) \cdot Pr(A/B_i)}{\sum_{j=1}^{n} Pr(B_j) \cdot Pr(A/B_j)} \tag{2}$$

Distribution Functions

- **Probability Mass Function** (PMF): Probability that a discrete random variable (vector) is exactly equal to some value in the sample space

$$p_Y(0) = Pr(Y = 0) \text{ and } p_Y(1) = Pr(Y = 1)$$

- **Probability Density Function** (PDF): Function that describes the relative likelihood for this random variable to occur at a given point in the sample space, that is $p(\phi_j(.) = v)$. And if $D \subseteq \Re$,

$$Pr(\phi_j(.) \in D) = \int_D p(v)dv$$

- **Joint Density Function:** In the case of continuous random vectors, if $p(\phi_1(.), \phi_2(.), \ldots, \phi_p(.))$ is a joint pdf and if $D \subseteq \Re^p$,

$$Pr(\phi(.) \in D) = \int \int \ldots \int_{\mathbf{v} \in D} p(\mathbf{v})d\mathbf{v}$$

Marginalization

- Marginal probability is the unconditional probability P(A) of the event A; that is, the probability of A, regardless of whether event B did or did not occur.
  Example:

$$p_{\phi_i(.)}(\widehat{v}) =$$
$$\int_{v_1} .. \int_{v_{i-1}} \int_{v_{i+1}} .. \int_{v_p} p(v_1, .., v_{i-1}, \widehat{v}, v_{i+1}, .., v_p) dv_1 .. dv_{i-1} dv_{i+1} .. dv_p$$

Conditional Density

If $\phi(.)$ and Y are two random variable then we can define the conditional probability density

1. of Y given $\phi(.)$, denoted $Y|\phi(.)$, *i.e.*, **Discrete Case** $p_{Y|\phi(.)}\left(Y = y|\phi(x)\right)$: **Discriminative Probabilistic Classifier**

   E.g: Logistic Regression (single node sigmoid NN) directly models $p_{Y|\phi(.)}\left(Y = 1|\phi(x)\right) = \frac{1}{1+e^{-(w)^T\phi(x)}}$. Then $p_{Y|\phi(.)}(Y = 0|\phi(x)) = ?$
   OR
2. of $\phi(.)$ given Y, denoted $\phi(.)|Y$, *i.e.*, **Continuous case** $p_{\phi(.)|Y}\left(\mathrm{v}|y\right)$: **Generative Probabilistic Classifier**

Thus...

Joint Probability Distribution

- $p_{\phi(.),Y}(\mathbf{v}, Y = y)$ or simply written as $p(\mathbf{v}, y)$
  - "Probability density at $\phi(.) = \mathbf{v}$ and $Y = y$"

Conditional Probability Distribution

- $p\left(Y = y|\phi(x)\right)$ VS. $p_{\phi(.)|Y}\left(\mathbf{v}|Y = y\right)$ (or simply $p\left(\mathbf{v}|y\right)$)
  - "Probability of $Y = y$ given $\phi(x)$" OR "Probability of $\phi(.) = \mathbf{v}$ given $Y = y$"

Rules of Probability

- Sum Rule (marginalization/ summing out)

$$p\left(\phi(x)\right) = \sum_{y'} p\left(y', \phi(x), \right)$$

- Bayes Rule: Gives a way a way of reversing conditional probabilities
  $p(\mathbf{v}|y) = \frac{p(y|\mathbf{v})p(\mathbf{v})}{p(y)} = \frac{p(y,\mathbf{v})p(\mathbf{v})}{\Sigma_{\mathbf{v}} p(y|\mathbf{v})p(\mathbf{v})}$

Thus: Conditional pmf/pdf

If $\phi(.)$ and Y are two random variables then we can define the conditional probability density

1. of Y given $\phi(.)$, denoted $Y|\phi(.)$, *i.e.*, **Discrete Case**
   - $p(y|\phi(x))$ is directly modeled and while you do not need to invoke Bayes rule, you only need to ensure the sum rule (pmfs sum to 1).
   - Logistic Regression (single node sigmoid NN) directly models $p(Y=1|\phi(x)) = \frac{1}{1+e^{-(w)^T\phi(x)}}$ and $p(Y=0|\phi(x)) = \frac{e^{-(w)^T\phi(x)}}{1+e^{-(w)^T\phi(x)}}$

   OR

2. of $\phi(.)$ given Y, denoted $\phi(.)|Y$, *i.e.*, **Continuous case**
   $p(\mathbf{v}|y) = \frac{p(y|\mathbf{v})p(\mathbf{v})}{\int_{\mathbf{v}'} p(y|\mathbf{v}')p(\mathbf{v}')}$

Training LR (Single node sigmoidal NN)

1. Estimator is a function of the dataset $\mathcal{D} = \left\{(\phi(x^{(1)}, y^{(1)}), (\phi(x^{(2)}, y^{(2)}), \ldots, (\phi(x^{(n)}, y^{(n)}))\right\}$ which is meant to approximate the parameter w.
2. Maximum Likelihood Estimator: Estimator $\widehat{w}$ that maximizes the likelihood $L(\mathcal{D}; w)$ of the data $\mathcal{D}$.
   - Assumes that all the instances $(\phi(x^{(1)}, y^{(1)}), (\phi(x^{(2)}, y^{(2)}), \ldots, (\phi(x^{(n)}, y^{(n)}))$ in $\mathcal{D}$ are all independent and identically distributed (iid)
   - Thus, Likelihood is the probability of $\mathcal{D}$ under iid assumption:

$$\hat{w} = \max_w L(\mathcal{D}, w) = \max_w \prod_{i=1}^{n} p(y^{(i)} | \phi(x^{(i)})) =$$

$$\max_w \prod_{i=1}^{n} \left(\frac{1}{1 + e^{-(w)^T \phi(x)}} \right)^{y^{(i)}} \left(\frac{e^{-(w)^T \phi(x)}}{1 + e^{-(w)^T \phi(x)}} \right)^{1 - y^{(i)}}$$

   - $\hat{w}$ is an estimator for w

Training LR (Single node sigmoidal NN)

1. Thus, Maximum Likelihood Estimator for w is
$$\hat{w} = \max_w L(\mathcal{D}, w) = \max_w \prod_{i=1}^{n} p(y^{(i)}|\phi(x^{(i)})) =$$
$$\max_w \prod_{i=1}^{n} \left(\frac{1}{1 + e^{-(w)^T\phi(x)}}\right)^{y^{(i)}} \left(\frac{e^{-(w)^T\phi(x)}}{1 + e^{-(w)^T\phi(x)}}\right)^{1-y^{(i)}}$$
2. $\hat{w}$ is an estimator for w
    1. To maximize the likelihood $P(\mathcal{D}; w)$ with respect to w, one can minimize the negative log-likelihood $E(w) = -logP(\mathcal{D}; w)$ with respect to w. Derive the expression for $E(w)$.
    2. $E(w)$ can be minimized with respect to w using gradient descend algorithm. Derive the expression of the gradient of $E(w)$ with respect to w.