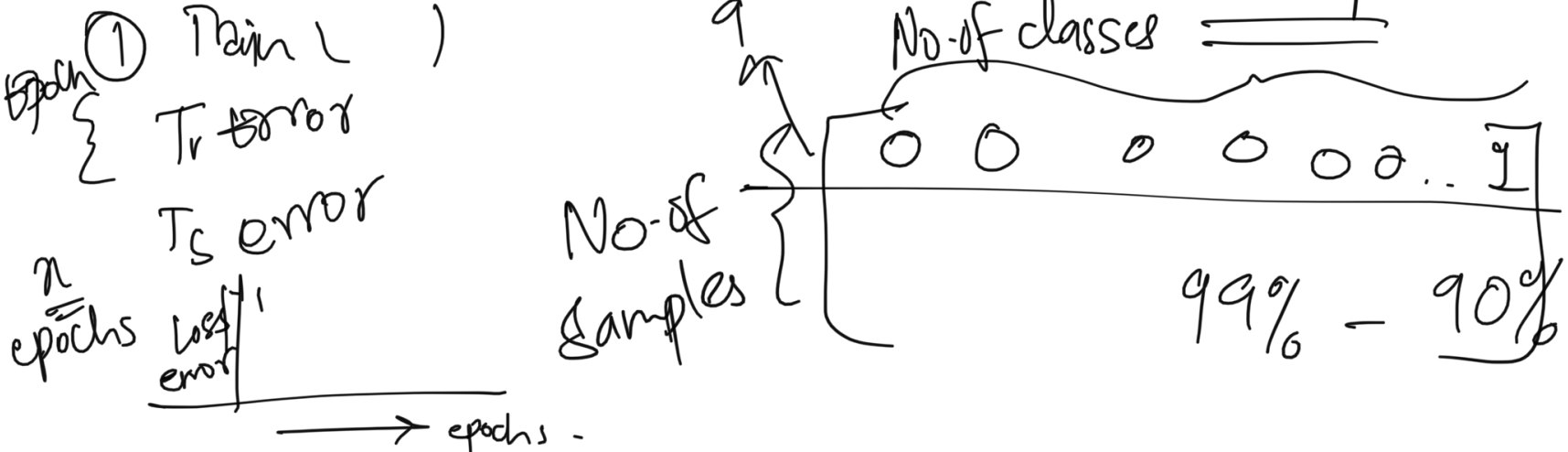
CSE 575: Statistical Machine Learning (Spring 2021)
Instructor: Nupur Thakur

Final Exam Review

Table of contents

1. Concepts Review
2. Sample Questions



Maximum Likelihood Estimation

- Given some training data and assuming a parametric model p(x|θ); what specific θ will fit/explain the data best?
- To consider all the samples denoted by D={x_1, x_2, ……, x_n}, assume that all the samples are i.i.d - independent and identically distributed.
- So, data likelihood represented by L(θ) is -

$$L(\boldsymbol{\theta}) = P(D|\boldsymbol{\theta}) = \prod_i P(x_i|\theta)$$

$$\widehat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta})$$

Naive Bayes

- The "naive" conditional independence assumption: each feature is (conditionally) independent of every other feature, given the label, i.e., $p(x_i \mid \{x_j \text{ for any } j \neq i\}, y) = p(x_i|y)$
- The predicted label is given by -

$$\hat{y} = \underset{y}{\operatorname{argmax}} P(y) \prod_{i=1}^{d} p(x_i|y)$$

Linear Regression

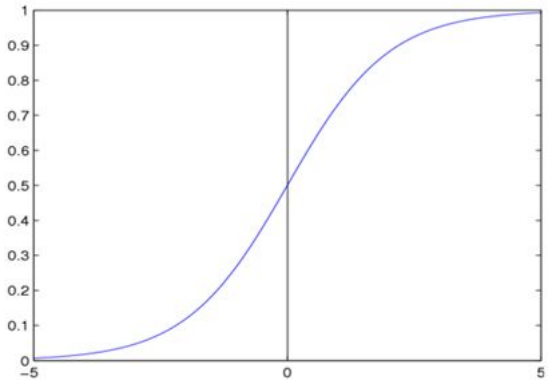
- Regression - A training set of n samples $<x^{(i)}, y^{(i)}>$ where $y^{(i)}$ is a continuous “label” (or target value) for $x^{(i)}$
- Linear regression - modeling the relation between y and x via a linear function $y \approx w_0 + w_1 x_1 + \ldots + w_d x_d = \mathbf{w}^t\mathbf{x}$
- The error is given as - $||e||^2 = ||y - X\,w||^2$

Logistic Regression

- Training set: n labelled samples <x(i), y(i)>
- Use the logistic function for modeling P(y|x), considering only the case of y∈{0,1}

$$P(y=0|\mathbf{x}) = \frac{1}{1+\exp\left(w_0+\sum_{i=1}^{d} w_i x_i\right)}$$

$$P(y=1|\mathbf{x}) = \frac{\exp\left(w_0+\sum_{i=1}^{d} w_i x_i\right)}{1+\exp\left(w_0+\sum_{i=1}^{d} w_i x_i\right)}$$



$$\sigma(t) = \frac{1}{1+e^{-t}} = \frac{e^t}{1+e^t}$$

Support Vector Machines (SVM)

- Key idea - To find the decision boundary such that the margin is maximized.
- Data - $\langle x^{(i)}, y^{(i)} \rangle$, $y^{(i)} \in \{-1,1\}$, $x^{(i)} \in \boldsymbol{R}^d$, for all i=1,…,n
- Plane equations-

- Margin -

Hidden Markov Model (HMM)

- Dynamic Bayesian network - modeling the process indexed by time.
- Two assumptions -
  - First-order Markov chain - $P(s^t=S_j \mid s^{t-1}=S_i)$
  - $a_{ij} = P(s^t=S_j \mid s^{t-1}=S_i)$, $1 \le i,j \le N$, for any t
- Specifying HMM - Θ, $\mathbf{\Omega}$, π, A, B

Problems in HMM

- For a given HMM $\Lambda=\{\Theta,\Omega,A,B,\pi\}$
  - Estimation of model parameters
  - Given an observation sequence **O**= $\{o^1,o^2, \dots ,o^k\}$, what is the most likely state sequence S = $\{s^1, s^2, \dots ,s^k\}$ that has produced O?
  - How likely is an observation O?

K-Means Clustering

Given: n samples, a number k.

Begin

initialize μ_1, μ_2, …, μ_k (randomly selected)

do classify n samples according to nearest μ_i

recompute μ_i

until no change in μ_i

return μ_1, μ_2, …, μ_k

End

Clusters

EM algorithm for GMM

Gaussian Mixture Models

Expectation Maximization

1. Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k, and evaluate the initial value of the log likelihood.
2. **E step**. Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{9.23}$$

3. **M step**. Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \tag{9.24}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})^{\mathrm{T}} \tag{9.25}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \tag{9.26}$$

where

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}). \tag{9.27}$$

4. Evaluate the log likelihood

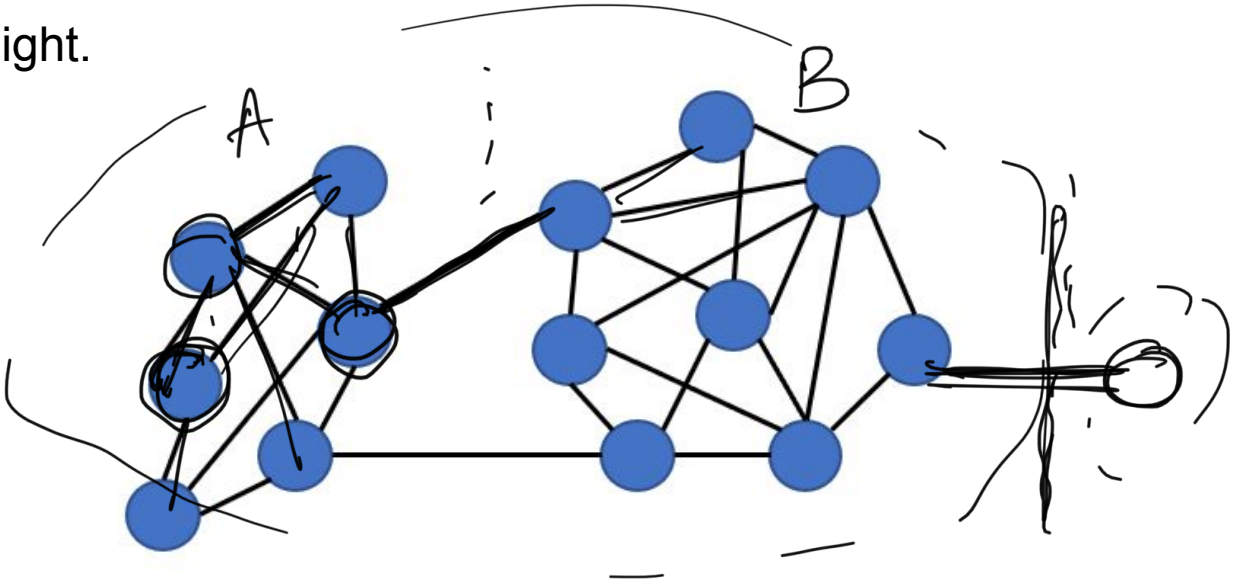
$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{9.28}$$

and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

= Bishop's book.

Clustering as a Graph Partition

- Find a partition of a graph such that the edges between different groups have a very low weight while the edges within a group have high weight.



Types of cuts

$$\text{Min cut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A}_i)$$

$$\text{RatioCut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{|A_i|} = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A}_i)}{|A_i|}$$

$|A_i|$ ← No. of nodes in cluster

$$\text{Ncut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{\text{vol}(A_i)} = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A}_i)}{\text{vol}(A_i)}.$$

$$\text{MinMaxCut}(A_1, \ldots, A_k) := \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A}_i)}{W(A_i, A_i)}.$$

PCA Algorithm

— Dimensionality Reduction.

d dimensions.

1. Compute the $d \times d$ sample covariance matrix C

2. Find the eigenvalues and corresponding eigenvectors of C

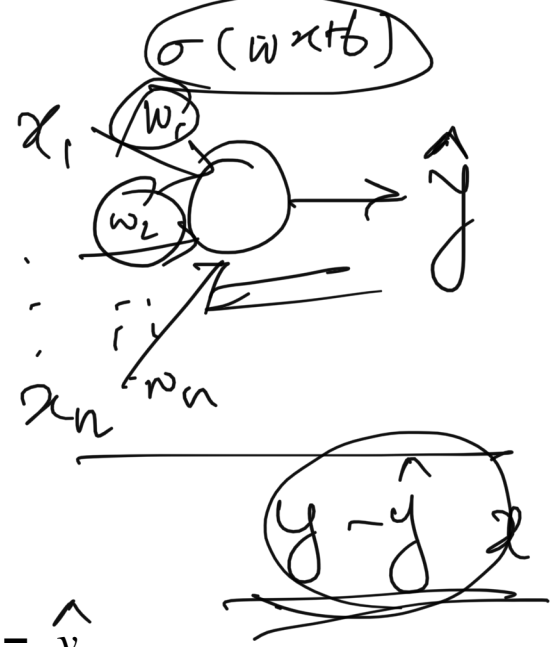
3. Project the original data onto the space spanned by the eigenvectors

   $d' \leq d$

   – The projection may be done onto a d'-dimensional subspace spanned by the first d' eigenvectors (ordered by the eigenvalue in descending order)

     • d' is determined by the desired accuracy

Learning in Perceptron



Iterate for t until a stop criterion is met

{

for each sample x_i with label y_i:

{

compute the output (y_i) of the network

estimate the error of the network $e(w(t))= y_i - \hat{y}_i$

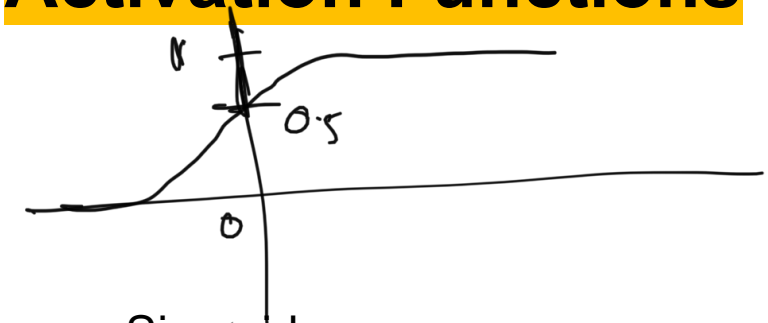
update the weight $w(t+1)=w(t)+e(w(t))x_i$

}

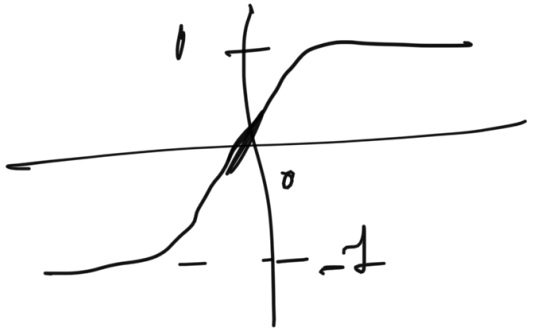
t++

}

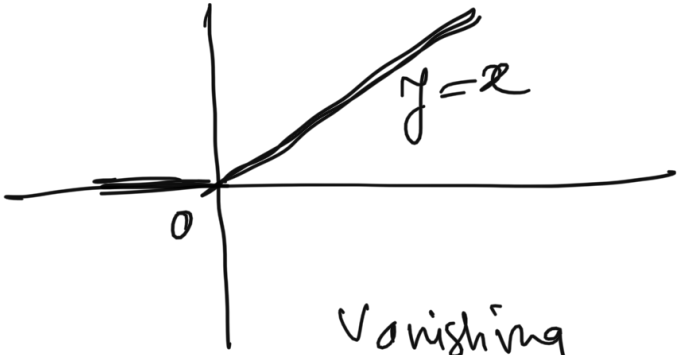
Activation Functions



Sigmoid

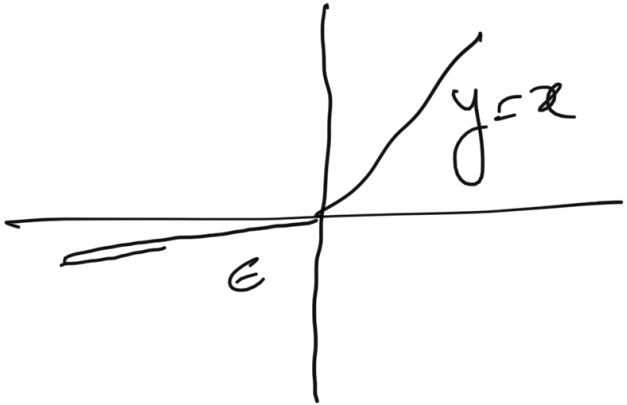


TanH



ReLU

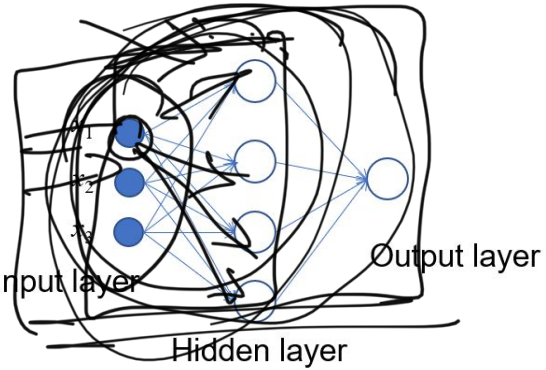
Vanishing gradients



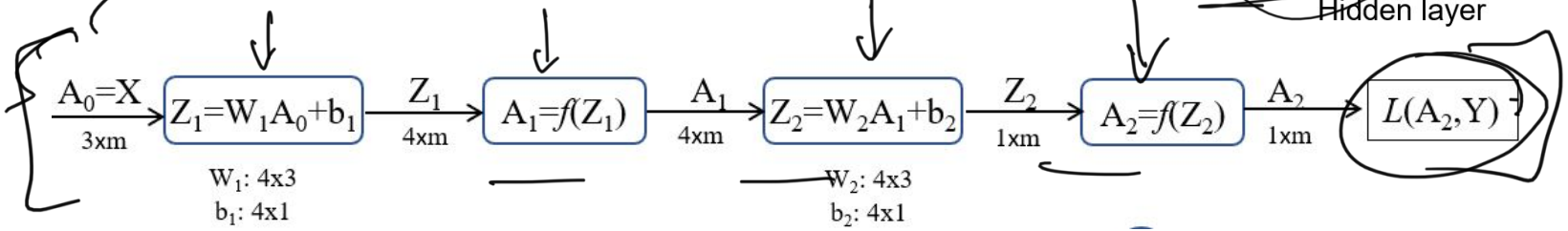
Leaky ReLU

Multi-layer perceptron

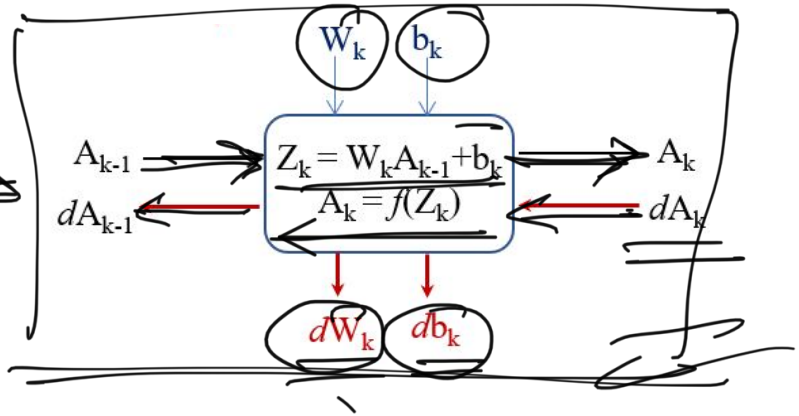
MLP Learning



Forward pass



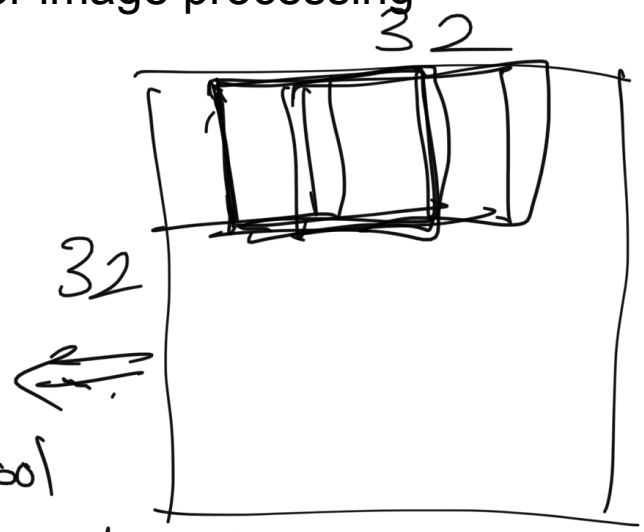
Backpropagation phase.



Convolutional Neural Network (CNN)

3X3X4

- Basic building block is the convolution for image processing
- Weight sharing - reduces parameters
- Learn the kernels based on the task.
- Layers -



convolutional layers.

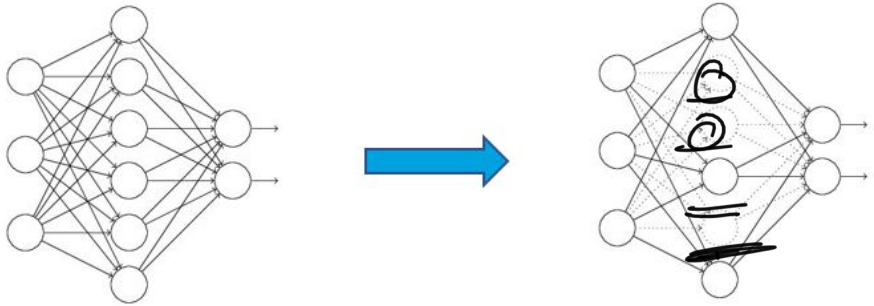
Pooling layer. — MaxPool

Activation layers - ReLU, Leaky ReLU.

Fully-connected — End of the network

Dropout

- Obtain (b) by randomly deactivate some hidden nodes in (a).
- Reducing co-adaptation of neurons



Batch Normalization

- Normalizes layer inputs of a batch

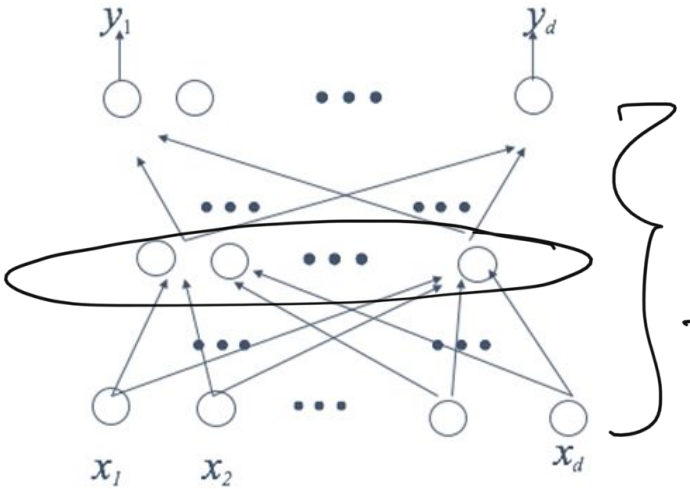
$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$$

Paper about BN: https://arxiv.org/pdf/1502.03167.pdf

Autoencoder — Unsupervised.

- Train a network without supervision
- y_i being an approximation of x_i



Questions?