ENGR-E 533 “Deep Learning Systems” Lecture 02: Baby Optimization

Minje Kim

Department of Intelligent Systems Engineering

Email: minje@indiana.edu

Website: http://minjekim.com

Research Group: http://saige.soic.indiana.edu

Meeting Request: http://doodle.com/minje

INDIANA UNIVERSITY

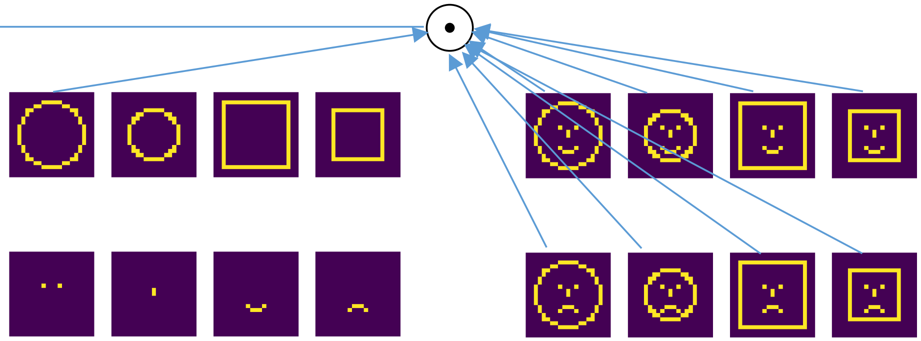
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

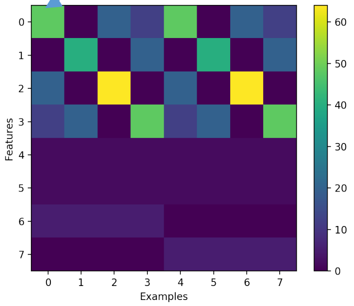
Baby Optimization

- How do we estimate the weights?

- Recall the feature extraction process

$$\sum_{i,j}$$



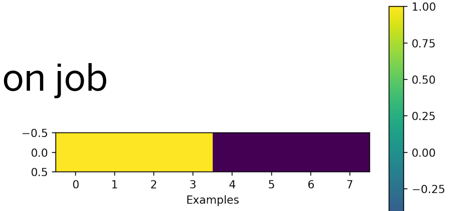


- Or,

$$\boldsymbol{H} = \boldsymbol{W}^\top \boldsymbol{X}$$

Baby Optimization

- The closed form solution



- Having $\boldsymbol{H}$ as the new input we can find another set of weights for the classification job
- Recall that I magically manually found out them

$$\boldsymbol{w}^{(2)} = [0,0,0,0,0,0,1,-1]^\top \qquad \hat{\boldsymbol{y}}^\top = \boldsymbol{w}^{(2)\top}\boldsymbol{H}$$

- If you choose to do optimization we have the pseudo inversed-based solution

$$\arg\max_{\boldsymbol{w}^{(2)}} \sum_t \mathcal{D}(y_t||\hat{y}_t) = \arg\max_{\boldsymbol{w}^{(2)}} \sum_t (y_t - \hat{y}_t)^2 = \arg\max_{\boldsymbol{w}^{(2)}} (\boldsymbol{y}^\top - \boldsymbol{w}^{(2)\top}\boldsymbol{H})(\boldsymbol{y}^\top - \boldsymbol{w}^{(2)\top}\boldsymbol{H})^\top$$

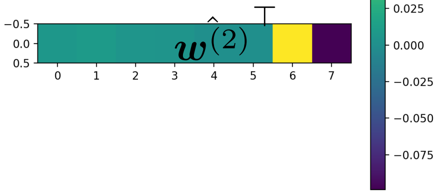
$$= \arg\max_{\boldsymbol{w}^{(2)}} (\boldsymbol{y}^\top\boldsymbol{y} + \boldsymbol{w}^{(2)\top}\boldsymbol{H}\boldsymbol{H}^\top\boldsymbol{w}^{(2)} - 2\boldsymbol{y}^\top\boldsymbol{H}^\top\boldsymbol{w}^{(2)})$$

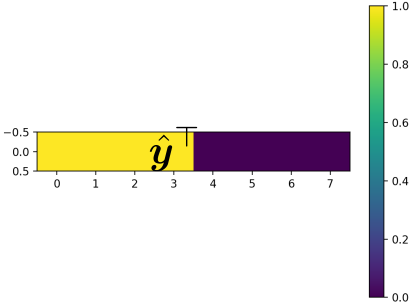
$$\frac{\partial \boldsymbol{y}^\top\boldsymbol{y} + \boldsymbol{w}^{(2)\top}\boldsymbol{H}\boldsymbol{H}^\top\boldsymbol{w}^{(2)} - 2\boldsymbol{y}^\top\boldsymbol{H}^\top\boldsymbol{w}^{(2)}}{\partial \boldsymbol{w}^{(2)}} = 2\boldsymbol{H}\boldsymbol{H}^\top\boldsymbol{w}^{(2)} - 2\boldsymbol{H}\boldsymbol{y}^\top = 0$$

$$\boldsymbol{w}^{(2)} = (\boldsymbol{H}\boldsymbol{H}^\top)^{-1}\boldsymbol{H}\boldsymbol{y}^\top$$

- I define my target output

$$\boldsymbol{y} = [1,1,1,1,0,0,0,0]^\top$$





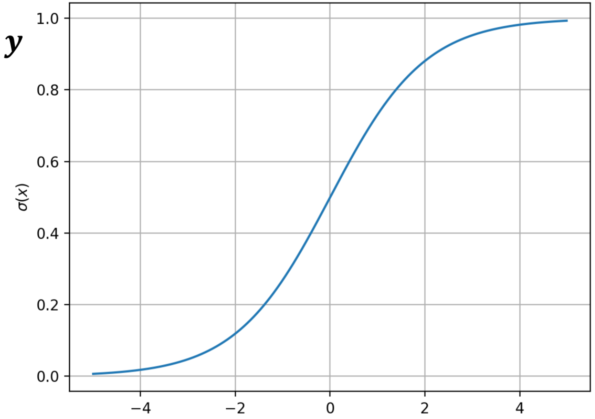
Baby Optimization

- Gradient descent

- Was it too easy?
  - Let me make it a little more difficult
- What I'm concerned is the range of $\hat{\boldsymbol{y}}$, the least mean squared error solution

$$\hat{\boldsymbol{y}}^\top = {\boldsymbol{w}^{(2)}}^\top \boldsymbol{H} \qquad \hat{\boldsymbol{y}} = \left((\boldsymbol{H}\boldsymbol{H}^\top)^{-1}\boldsymbol{H}\boldsymbol{y}^\top\right)^\top \boldsymbol{H}$$

  - There's no guarantee that $\hat{\boldsymbol{y}}$ is between 0 and 1
- Why BETWEEN 0 and 1?
  - We care about the probability $P(\text{Class A}|\boldsymbol{X}_{:,t})$
  - If you're absolutely sure, then this value will be 1 or 0 as in the target variable $\boldsymbol{y}$
  - In other words, your training samples are with absolutely sure class labels
  - While for your test samples you need to predict the labels by using a posterior probability of the label given the data point
- Any idea?
  - Hint: You Already Learned This (YALT)
  - Logistic function!



Baby Optimization

- Gradient descent

- So, we wrap the output values with the logistic sigmoid function

$$\boldsymbol{z}^\top = \boldsymbol{w}^{(2)^\top} \boldsymbol{H} \qquad \hat{\boldsymbol{y}}^\top = \sigma\left(\boldsymbol{z}^\top\right)$$

Note that the sigmoid function works element-wise

- Closed form solution?

$$\arg\max_{\boldsymbol{w}^{(2)}} \sum_t (y_t - \hat{y}_t)^2 = \arg\max_{\boldsymbol{w}^{(2)}} \left(\boldsymbol{y}^\top \boldsymbol{y} + \sigma\left(\boldsymbol{w}^{(2)^\top} \boldsymbol{H}\right) \sigma\left(\boldsymbol{H}^\top \boldsymbol{w}^{(2)}\right) - 2\boldsymbol{y}^\top \sigma\left(\boldsymbol{H}^\top \boldsymbol{w}^{(2)}\right)\right)$$

  - Maybe not a quadratic function anymore
- What does this mean (I mean 'no closed form solution')?
  - There can be many stationary points
  - Impossible to find the global optimum
  - Welcome to the machine learning world!
- Instead, we do numerical optimization
  - We start from randomly initialized parameters
  - Check the error using the current parameters
  - Find the negative gradient direction that reduce the error
  - Update the parameters using the negative gradient direction

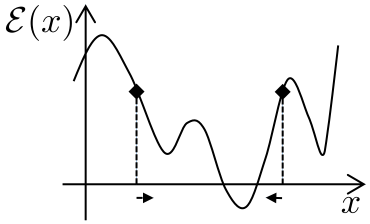
Baby Optimization
- Gradient descent

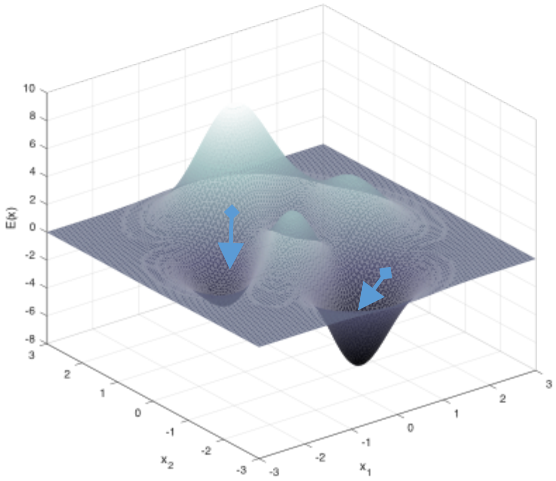
- i=0
  1. Initialize parameters with small random numbers
  2. Calculate the output using all training samples (i.e. input and target pairs)
     $$\boldsymbol{z}^\top = \boldsymbol{w}^{(2)\top}\boldsymbol{H} \qquad \hat{\boldsymbol{y}}^\top = \sigma\left(\boldsymbol{z}^\top\right)$$
  3. Calculate the error (cost)
     $$\mathcal{E} = \sum_t (y_t - \hat{y}_t)^2$$
  4. Update the parameters $\boldsymbol{w}^{(2)} \leftarrow \boldsymbol{w}^{(2)} - \boxed{\rho}\boxed{\frac{\partial \mathcal{E}}{\partial \boldsymbol{w}^{(2)}}}$

Learning rate: how much to move?

Gradient: in which direction?

- i>0
  - Repeat 2-4
- You should ask two questions
  - What is ρ?
    - Learning rate
  - How do we calculate $\frac{\partial \mathcal{E}}{\partial \boldsymbol{w}^{(2)}}$?
    - It depends on the problem
    - But basically by doing differentiation





Baby Optimization

- Gradient descent

$$z_t = \boldsymbol{w}^{(2)\top} \boldsymbol{H}_{:,t}$$

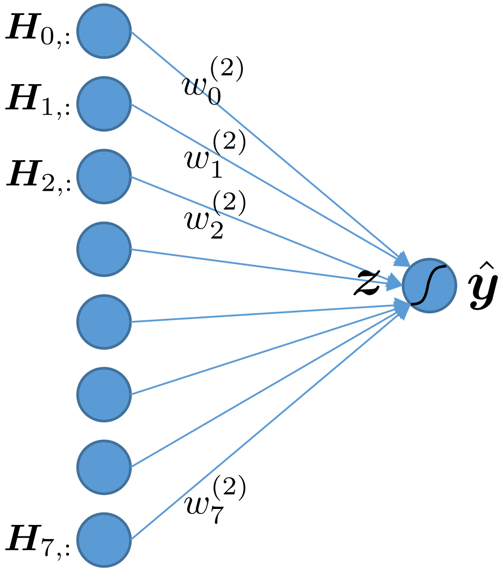
$$\hat{y}_t = \sigma(z_t)$$

$$\mathcal{E} = \sum_t (y_t - \hat{y}_t)^2$$

- So, let's do the differentiation
- The chain rule

$$\frac{\partial \mathcal{E}}{\partial \boldsymbol{w}^{(2)}} = \sum_t \frac{\partial \mathcal{E}}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial z_t} \frac{\partial z_t}{\partial \boldsymbol{w}^{(2)}} = \sum_t 2(\hat{y}_t - y_t)\, \sigma'(z_t)\, \boldsymbol{H}_{:,t}$$

$$= \boldsymbol{H} \underbrace{\big((2\hat{\boldsymbol{y}} - 2\boldsymbol{y}) \odot \sigma'(\boldsymbol{z})\big)}_{\text{BP error}}$$



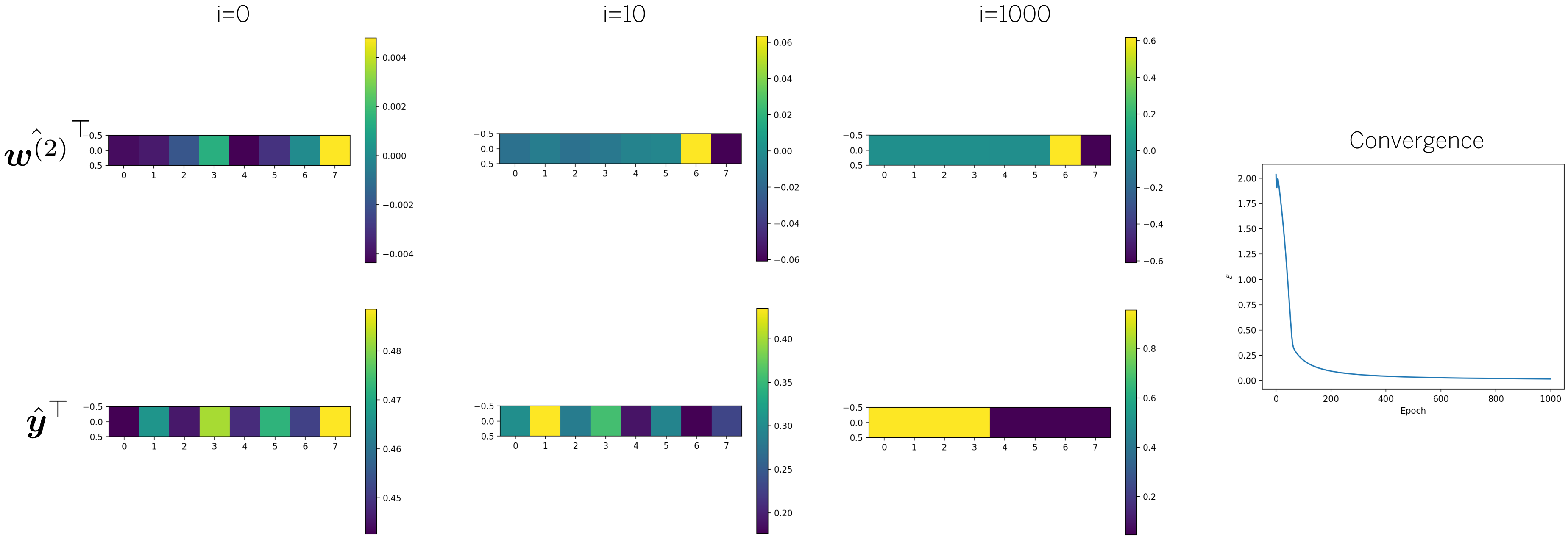
- The gradient direction for the weights is defined by
  - The inner product of the data matrix and the **backpropagation error**
- Backpropagation error?
  - Partial derivative of the total cost w.r.t. the input to a node
  - In this case, $\frac{\partial \mathcal{E}}{\partial z_t}$

Google "matrix cookbook"

Baby Optimization

- Gradient descent

- Parameters and predictions over epochs



Baby Optimization

- Gradient descent: softmax

- Softmax regression is special
  - So is logistic regression
  - The output vector sums to one and nonnegative
    - A probabilistic distribution over classes
- Sum of squared error might not be the best
- Then, why not using a more suitable error metric?
  - Cross entropy $-\sum_i q(x_i)\log p(x_i)$

$$\mathcal{E} = -\sum_t \sum_c \mathbf{Y}_{c,t} \log \hat{\mathbf{Y}}_{c,t}$$

- Eventually, we want a partial differentiation of $\mathcal{E}$, but it's not that simple

$$\frac{\partial \mathcal{E}}{\partial \boldsymbol{W}^{(2)}_{j,:}} = \sum_t \sum_c \frac{\partial \mathcal{E}}{\partial \hat{\mathbf{Y}}_{c,t}} \frac{\partial \hat{\mathbf{Y}}_{c,t}}{\partial \boldsymbol{Z}_{j,t}} \frac{\partial \boldsymbol{Z}_{j,t}}{\partial \boldsymbol{W}^{(2)}_{j,:}}$$

Note: j and c are the same index for the three output units, but j is for the input to the units and c is for the output. For example, $\boldsymbol{W}^{(2)}_{0,:}$ is for the green arrows

$$\frac{\partial \mathcal{E}}{\partial \hat{\mathbf{Y}}_{c,t}} = -\frac{\mathbf{Y}_{c,t}}{\hat{\mathbf{Y}}_{c,t}} \qquad \frac{\partial \hat{\mathbf{Y}}_{c,t}}{\partial \boldsymbol{Z}_{j,t}} = ? \qquad \frac{\partial \boldsymbol{Z}_{j,t}}{\partial \boldsymbol{W}^{(2)}_{j,:}} = \boldsymbol{H}^\top_{:,t}$$

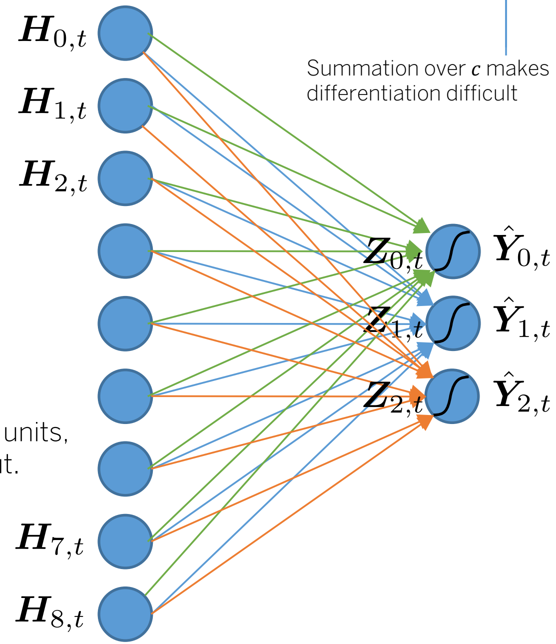
$$P(\text{Class } \#0|\boldsymbol{X}_{:,t}) = \frac{\exp(\boldsymbol{Z}_{0,t})}{\sum_{c=0}^{\mathcal{C}-1}\exp(\boldsymbol{Z}_{c,t})}$$

$$P(\text{Class } \#1|\boldsymbol{X}_{:,t}) = \frac{\exp(\boldsymbol{Z}_{1,t})}{\sum_{c=0}^{\mathcal{C}-1}\exp(\boldsymbol{Z}_{c,t})}$$

$$\cdots$$

$$P(\text{Class } \#\mathcal{C}-1|\boldsymbol{X}_{:,t}) = \frac{\exp(\boldsymbol{Z}_{\mathcal{C}-1,t})}{\sum_{c=0}^{\mathcal{C}-1}\exp(\boldsymbol{Z}_{c,t})}$$

Summation over c makes differentiation difficult



Baby Optimization

- Gradient descent: softmax

$$P(\text{Class } \#0|\boldsymbol{X}_{:,t}) = \hat{\boldsymbol{Y}}_{0,t} = \frac{\exp(\boldsymbol{Z}_{0,t})}{\sum_{c=0}^{\mathcal{C}-1}\exp(\boldsymbol{Z}_{c,t})}$$

$$P(\text{Class } \#1|\boldsymbol{X}_{:,t}) = \hat{\boldsymbol{Y}}_{1,t} = \frac{\exp(\boldsymbol{Z}_{1,t})}{\sum_{c=0}^{\mathcal{C}-1}\exp(\boldsymbol{Z}_{c,t})}$$

$$\cdots$$

$$P(\text{Class } \#\mathcal{C}-1|\boldsymbol{X}_{:,t}) = \hat{\boldsymbol{Y}}_{\mathcal{C}-1,t} = \frac{\exp(\boldsymbol{Z}_{\mathcal{C}-1,t})}{\sum_{c=0}^{\mathcal{C}-1}\exp(\boldsymbol{Z}_{c,t})}$$

- Any $\widehat{\boldsymbol{Y}}_{c,t}$ involves all $\boldsymbol{Z}_{j,t}\ \forall j$

$$\frac{\partial \hat{\boldsymbol{Y}}_{c,t}}{\partial \boldsymbol{Z}_{j,t}} = \frac{\exp(\boldsymbol{Z}_{c,t})\left(\sum_i \exp(\boldsymbol{Z}_{i,t})\right) - \left(\exp(\boldsymbol{Z}_{c,t})\right)^2}{\left(\sum_i \exp(\boldsymbol{Z}_{i,t})\right)^2} \qquad \text{if } j = c$$

$$= \hat{\boldsymbol{Y}}_{c,t}(1-\hat{\boldsymbol{Y}}_{c,t}) = \hat{\boldsymbol{Y}}_{j,t}(1-\hat{\boldsymbol{Y}}_{j,t})$$

$$\frac{\partial \hat{\boldsymbol{Y}}_{c,t}}{\partial \boldsymbol{Z}_{j,t}} = -\frac{\exp(\boldsymbol{Z}_{c,t})\exp(\boldsymbol{Z}_{j,t})}{\left(\sum_i \exp(\boldsymbol{Z}_{i,t})\right)^2} \qquad \text{if } j \neq c$$

$$= -\hat{\boldsymbol{Y}}_{c,t}\hat{\boldsymbol{Y}}_{j,t}$$

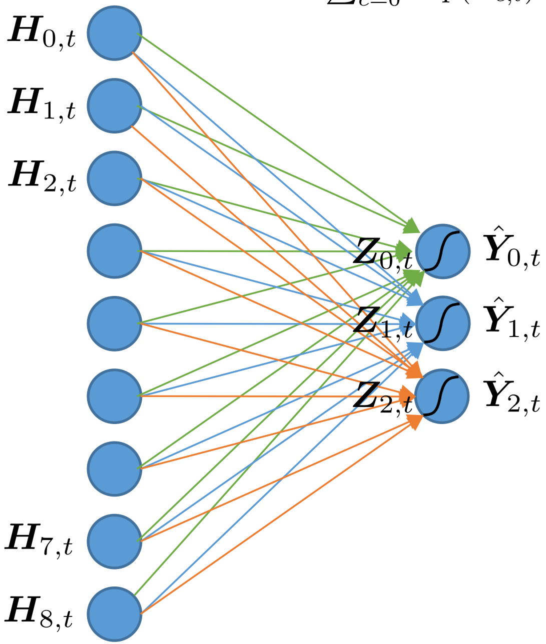
$$\frac{\partial \mathcal{E}}{\partial \boldsymbol{W}^{(2)}_{j,:}} = \sum_t \sum_c \frac{\partial \mathcal{E}}{\partial \hat{\boldsymbol{Y}}_{c,t}} \frac{\partial \hat{\boldsymbol{Y}}_{c,t}}{\partial \boldsymbol{Z}_{j,t}} \frac{\partial \boldsymbol{Z}_{j,t}}{\partial \boldsymbol{W}^{(2)}_{j,:}}$$

$$= \sum_t \left(-\frac{\boldsymbol{Y}_{j,t}}{\hat{\boldsymbol{Y}}_{j,t}} \hat{\boldsymbol{Y}}_{j,t}(1-\hat{\boldsymbol{Y}}_{j,t}) + \sum_{c\neq j} \frac{\boldsymbol{Y}_{c,t}}{\hat{\boldsymbol{Y}}_{c,t}} \hat{\boldsymbol{Y}}_{c,t}\hat{\boldsymbol{Y}}_{j,t} \right) \boldsymbol{H}_{:,t}^\top$$

$$= \sum_t \left(-\boldsymbol{Y}_{j,t}(1-\hat{\boldsymbol{Y}}_{j,t}) + \sum_{c\neq j} \boldsymbol{Y}_{c,t}\hat{\boldsymbol{Y}}_{j,t} \right) \boldsymbol{H}_{:,t}^\top$$

$$= \sum_t \left(-\boldsymbol{Y}_{j,t} + \sum_c \boldsymbol{Y}_{c,t}\hat{\boldsymbol{Y}}_{j,t} \right) \boldsymbol{H}_{:,t}^\top = (\hat{\boldsymbol{Y}}_{j,:} - \boldsymbol{Y}_{j,:})\boldsymbol{H}^\top$$

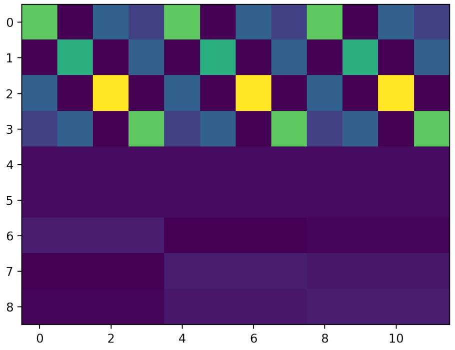
$$\boxed{\frac{\partial \mathcal{E}}{\partial \boldsymbol{W}^{(2)}} = (\hat{\boldsymbol{Y}} - \boldsymbol{Y})\boldsymbol{H}^\top}$$



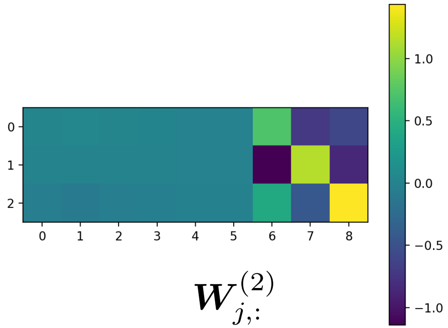
Baby Optimization

- Gradient descent: softmax

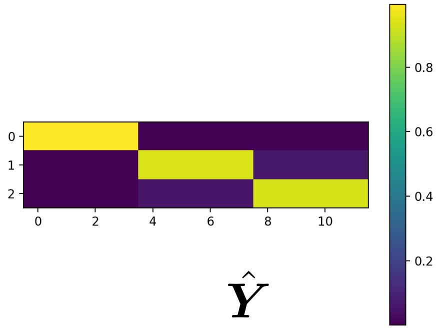
- Parameter estimates: the three-class case



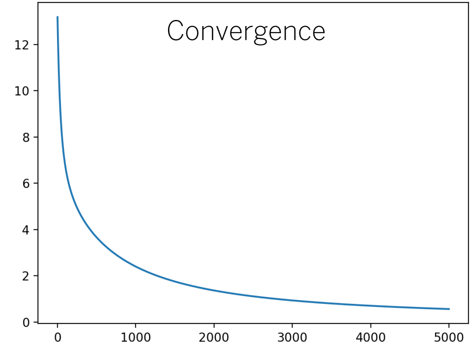
$$\boldsymbol{H} = \boldsymbol{W}^\top \boldsymbol{X}$$



$$\boldsymbol{W}_{j,:}^{(2)}$$



$$\hat{\boldsymbol{Y}}$$



Take-home Messages

- Parameter estimation for nonlinear models involve complicated optimization procedure
  - No global optimum
  - Differentiation
  - Learning rate
  - Gradient directions
  - Initialization
  - Computational cost
- In the last layer the choice of the activation function and cost function matters
  - You want to transform the output variable so that
    - Its dynamic range matches the target variable
  - The cost function needs to be chosen accordingly

Thank You!

Minje Kim
Department of Intelligent Systems Engineering
Email: minje@indiana.edu
Website: http://minjekim.com
Research Group: http://saige.soic.indiana.edu
Meeting Request: http://doodle.com/minje