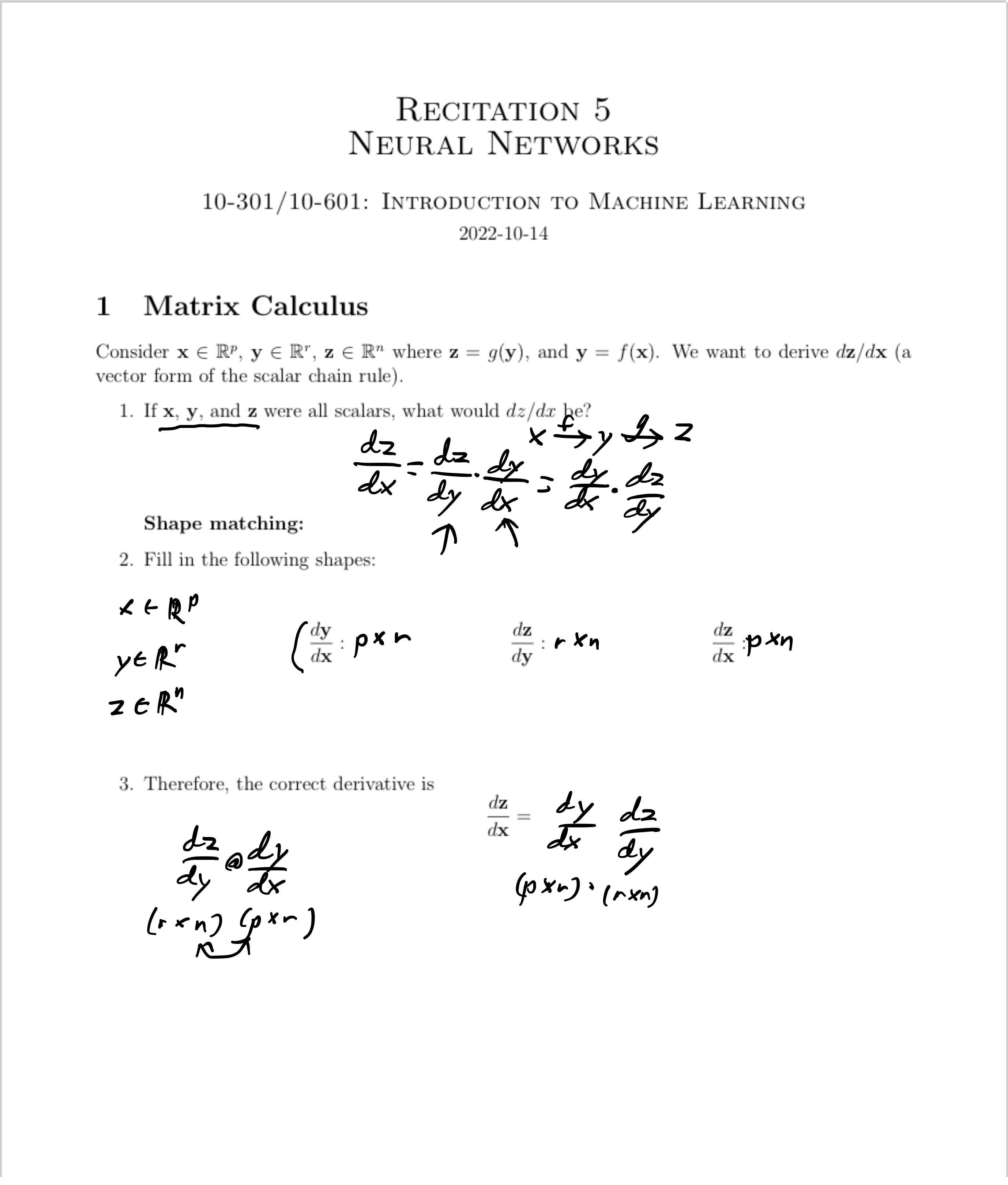
HW5 Recitation

Wednesday, March 1, 2023 9:24 AM



Recitation 5
Neural Networks

10-301/10-601: Introduction to Machine Learning

2022-10-14

1 Matrix Calculus

Consider $\mathbf{x} \in \mathbb{R}^p$, $\mathbf{y} \in \mathbb{R}^r$, $\mathbf{z} \in \mathbb{R}^n$ where $\mathbf{z} = g(\mathbf{y})$, and $\mathbf{y} = f(\mathbf{x})$. We want to derive $d\mathbf{z}/d\mathbf{x}$ (a vector form of the scalar chain rule).

1. If $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ were all scalars, what would dz/dx be?

$$x \xrightarrow{f} y \xrightarrow{g} z$$

$$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx} = \frac{dy}{dx} \cdot \frac{dz}{dy}$$

Shape matching:

2. Fill in the following shapes:

$x \in \mathbb{R}^p$

$y \in \mathbb{R}^r$

$z \in \mathbb{R}^n$

$\frac{d\mathbf{y}}{d\mathbf{x}}$: $p \times r$ $\frac{d\mathbf{z}}{d\mathbf{y}}$: $r \times n$ $\frac{d\mathbf{z}}{d\mathbf{x}}$: $p \times n$

3. Therefore, the correct derivative is

$$\frac{d\mathbf{z}}{d\mathbf{x}} = \frac{dy}{dx} \frac{dz}{dy}$$

$$(p \times r) \cdot (r \times n)$$

$$\frac{dz}{dy} @ \frac{dy}{dx}$$

$$(r \times n) \; (p \times r)$$

Generalizing a single element: In order to ensure your derivatives are correct, we recommend you use matrix calculus rules whenever possible. When you're not sure how to apply a rule or if one applies, use the method of generalizing a single element.

4. Example: Suppose $\mathbf{A} \in \mathbb{R}^{n\times n}, \mathbf{x} \in \mathbb{R}^n$, and we want to compute $\frac{d\mathbf{Ax}}{d\mathbf{x}}$. Note that our numerator and denominator are both length-n vectors, so our derivative has shape $\mathbb{R}^{n\times n}$.

5. Is $\left(\frac{d\mathbf{Ax}}{d\mathbf{x}}\right)_{ij}$ equal to $\frac{d\mathbf{Ax}_i}{d\mathbf{x}_j}$ or $\frac{d\mathbf{Ax}_j}{d\mathbf{x}_i}$? Why?

$$\frac{dAx}{dx} \leftarrow n \qquad \uparrow n$$

$$\left(\frac{dAx}{dx}\right)_{ij} = \cancel{\frac{d(Ax)_i}{dx_j}} \text{ or } \boxed{\frac{d(Ax)_j}{dx_i}}$$

6. Compute $\left(\frac{d\mathbf{Ax}}{d\mathbf{x}}\right)_{ij}$:

$$\left(\frac{dAx}{dx}\right)_{ij} = \frac{d(Ax)_j}{dx_i}$$

$$= \cancel{dA_j \cdot x} \quad \frac{dA_j^T x}{dx_i}$$

$$= \frac{d}{dx_i} \sum_{k=1}^{n} A_{jk} x_k$$

$$= A_{ji}$$

$$A \qquad \boxed{A^T} \qquad \frac{dAx}{dx}$$

Applying Matrix Calculus For example, suppose we are finding the closed-form solution to linear regression: given $\mathbf{X} \in \mathbb{R}^{n\times d}, \mathbf{y} \in \mathbb{R}^n$, we wish to find $\boldsymbol{\theta} \in \mathbb{R}^d$ that minimizes $\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2$. ←

7. What is the shape of $\mathbf{X}^\top y$? Is this our solution? ← $d\times 1$ X^Ty is not the right soln

8. What is the closed-form solution? Use matrix calculus to derive the solution.

$$\underbrace{(X^TX)^{-1}}\underbrace{X^Ty}$$

$$\frac{d}{d\theta}\|y-X\theta\|_2^2 = \frac{d}{d\theta}(y-X\theta)^T(y-X\theta)$$

$$= \frac{d}{d\theta}(\cancel{y^Ty} - y^T(X\theta) - (X\theta)^Ty + (X\theta)^T(X\theta))$$

$$= \frac{d}{d\theta} -2y^TX\theta + (X\theta)^T(X\theta)$$

$$-2\frac{d}{d\theta}y^T(X\theta)$$

$$-2\frac{d}{d\theta}(y^TX)\theta$$

$$-2\frac{d}{d\theta}(X^Ty)^T\theta$$

$n\times 1$ $n\times 1$ $n\times 1$

$$-2X^Ty$$

$$= -2X^Ty + 2X^TX\theta$$

$$0 = -2X^Ty + 2X^TX\theta$$

$$X^Ty = X^TX\theta$$

$$(X^TX)^{-1}X^Ty = \theta$$

$$\frac{d}{d\theta}(X\theta)^T(X\theta)$$

$$g'(x)\cdot f'(g(x))$$

$$\left(\frac{d}{d\theta}X\theta\right)(2X\theta)$$

$$X^T2X\theta$$

2 Forward Propagation

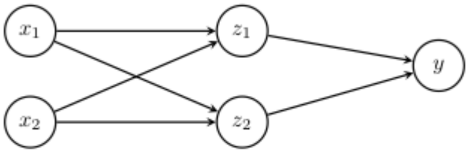


Figure 1: Neural Network For Example Questions

Forward Propagation is the process of calculating the value of your loss function, given data, weights and activation functions. Given the input data $\mathbf{x}$, we can transform it by the given weights, $\boldsymbol{\alpha}$, then apply the corresponding activation function to it and finally pass the result to the next layer. Forward propagation does not involve taking derivatives and proceeds from the input layer to the output layer.

Network Overview Consider the neural network with one hidden layer shown in Figure 2. The input layer consists of 2 features $\mathbf{x} = [x_1, x_2]^T$, the hidden layer has 2 nodes with output $\mathbf{z} = [z_1, z_2]^T$, and the output layer is a scalar $\hat{y}$. We also add an intercept to the input, $x_0 = 1$ and the output of the hidden layer $z_0 = 1$, both of which are fixed to 1.

$\boldsymbol{\alpha}$ is the matrix of weights from the inputs to the hidden layer and $\boldsymbol{\beta}$ is the matrix of weights from the hidden layer to the output layer. $\alpha_{j,i}$ represents the weight going *to* the node z_j in the hidden layer *from* the node x_i in the input layer (e.g. $\alpha_{1,2}$ is the weight from x_2 to z_1), and $\boldsymbol{\beta}$ is defined similarly. We will use a **ReLU** activation function for the hidden layer and no activation for the output layer.

Network Details Equivalently, we define each of the following.

The input:

$\overset{1}{\alpha_{j,0}}\ \overset{1}{x_0} + \dots$

$\alpha_{j,0} + \Sigma$

$$\mathbf{x} = [\overset{1}{x_0}, x_1, x_2]^T \tag{1}$$

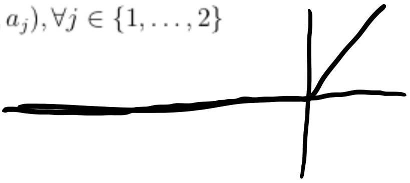
Linear combination at the first (hidden) layer:

$$\underbrace{a_j = \sum_{i=0}^{2} \alpha_{j,i} \cdot x_i}, \ \forall j \in \{1, \dots, 2\} \tag{2}$$

Activation at the first (hidden) layer:

$$z_j = \text{ReLU}(a_j) = \max(0, a_j), \forall j \in \{1, \dots, 2\} \tag{3}$$

$$\mathbf{z} = [z_0, z_1, z_2]^T \tag{4}$$



Linear combination at the second (output) layer:

$$\hat{y} = \sum_{i=0}^{2} \beta_j \cdot z_j, \tag{5}$$

Here we fold in the intercept term $\alpha_{j,0}$ by thinking of $x_0 = 1$, and fold in β_0 by thinking of $z_0 = 1$.

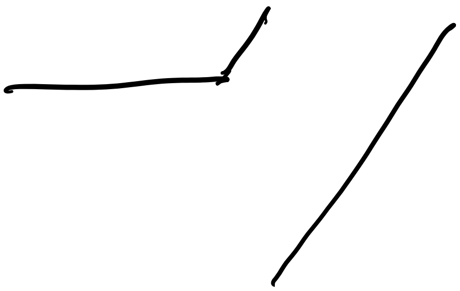
Loss We will use Squared error loss, $\ell(\hat{y}, y)$:

$$\ell(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2 \tag{6}$$

1. Why and how do we include an intercept term in the input and in the hidden-layer?

To encode shifts from the origin to help better fit data

2. Why do we need to use nonlinear activation functions in our neural net?



to prevent it from being linear regression

$a = (W_1 x)$

$b = W_2 (a) = W_2 \overset{\downarrow}{f}(W_1(x))$

$= \underbrace{(W_2 W_1)} x$

We initialize the network weights as:

$$\alpha = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

$$\beta = \begin{bmatrix} 0 & 1 & 2 \end{bmatrix}$$

For the following questions, we use $y = 3$.

1. **Scalar Form:**
   - Given $x_1 = 1$, $x_2 = 2$, What are the values of a?

$$a_2 = \sum_{i=0}^{2} \alpha_{2,i} x_i = 1 \cdot 0 + 1 \cdot 2 + 2 \cdot 0 = 2$$

$x = [1, 1, 2]$

   - Given $z_1 = 0, z_2 = 1$ calculate $\hat{y}, l$

$z = [1, 0, 1]$

$$\hat{y} = \sum_{i=0}^{2} \beta_i \cdot z_i = 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 2 = 2$$

$$\frac{1}{2}(2-3)^2 = \frac{1}{2}$$

2. **Vector Form:** Find the vector form of forward computation, given **x** is a column vector.

$\alpha \in \mathbb{R}^{2\times 3}$ $x \in \mathbb{R}^3$ $a \in \mathbb{R}^2$

$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \rightarrow a = \alpha x$

$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \rightarrow z = \mathrm{ReLU}(a)$

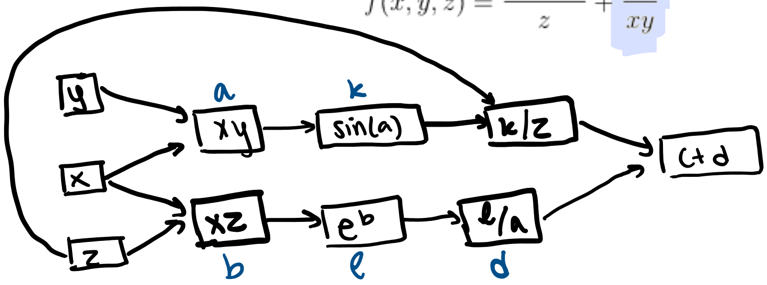
$\begin{bmatrix} z_0 = 1 \\ z \end{bmatrix}$ $\hat{y} = \beta z$

z_2

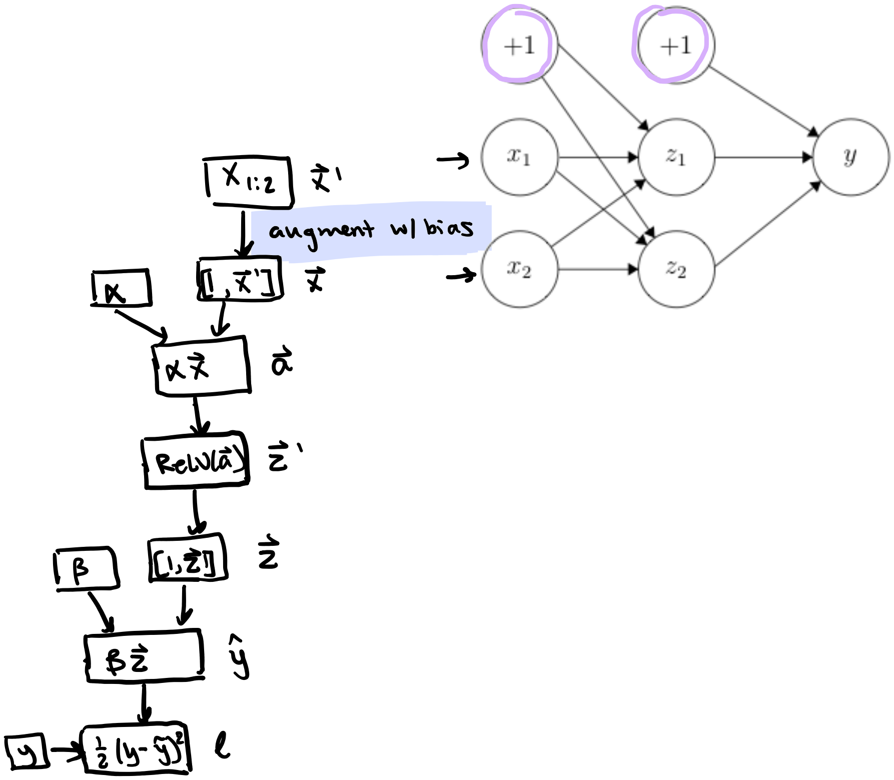
3 Computation Diagrams

1. For the following function f, create the computation graph using the conventions defined in lecture.

$$f(x, y, z) = \frac{\sin(xy)}{z} + \frac{e^{xz}}{xy}$$



2. For the following neural network, draw the corresponding computation graph. Assume that all hidden units use the ReLU function as the activation function and that the loss is mean squared error. Provide the shape of all parameters defined in the computation graph. Assume the weights for the first layer and second layers are respectively the matrices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.



4 Backward Propagation

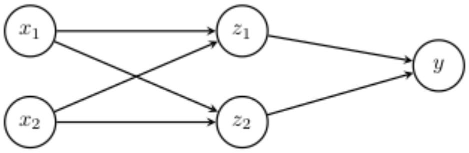


Figure 2: Neural Network For Example Questions

Given a Neural Network and a corresponding loss function $J(\theta)$, backpropagation gives us the gradient of the loss function with respect to the weights of the neural network. The method is called *backward* propagation because we calculate the gradients of the final layer of weights first, then proceed backward to the first layer. In a simple neural network with one hidden layer, the partial derivatives that we need for learning are $\frac{\partial \ell}{\partial \alpha_{ij}}$ and $\frac{\partial \ell}{\partial \beta_{kj}}$, and we need to apply chain rule recursively to obtain these. Note that in implementation, it is easier to use matrix/vector forms to conduct computations.

1. Many gradients are calculated in back propagation. Which of these gradients are used to update the weights? Do not include intermediate value(s) used to calculate these gradient(s).

the gradients used to update the weights are the gradients of the loss func w/ respect to α and β $(\partial \ell / \partial \alpha, \partial \ell / \partial \beta)$

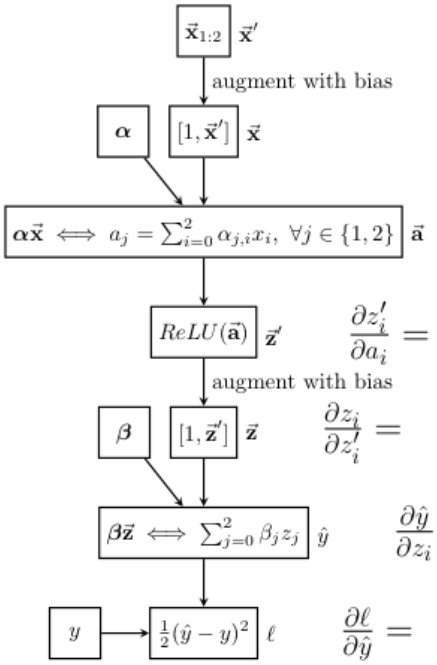
2. **Scalar Form:** Given

- $x_1 = 1$, $x_2 = 2$
- $a_1 = 3$, $a_2 = 2$
- $z_1 = 3$, $z_2 = 2$
- $\boldsymbol{\alpha} = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \end{bmatrix}$
- $\boldsymbol{\beta} = \begin{bmatrix} 0 & 1 & 2 \end{bmatrix}$
- $y = 3$,

what are the values of $\frac{\partial \ell}{\partial \beta_1}$, $\frac{\partial \ell}{\partial \alpha_{1,1}}$?

Hint: Derive expressions for $\frac{\partial \ell}{\partial \beta_i}$ and $\frac{\partial \ell}{\partial \alpha_{i,j}}$ first, then substitute in values.

For convenience, the computation graph for the neural network is displayed below:



$\vec{\mathbf{x}}_{1:2}$ $\vec{\mathbf{x}}'$
augment with bias
$\boldsymbol{\alpha}$
$[1, \vec{\mathbf{x}}']$ $\vec{\mathbf{x}}$
$\boldsymbol{\alpha}\vec{\mathbf{x}} \iff a_j = \sum_{i=0}^{2} \alpha_{j,i} x_i, \ \forall j \in \{1, 2\}$ $\vec{\mathbf{a}}$
$ReLU(\vec{\mathbf{a}})$ $\vec{\mathbf{z}}'$
augment with bias
$\boldsymbol{\beta}$
$[1, \vec{\mathbf{z}}']$ $\vec{\mathbf{z}}$
$\boldsymbol{\beta}\vec{\mathbf{z}} \iff \sum_{j=0}^{2} \beta_j z_j$ $\hat{y}$
y
$\frac{1}{2}(\hat{y} - y)^2$ ℓ

$\frac{\partial z_i'}{\partial a_i} =$

$\frac{\partial z_i}{\partial z_i'} =$

$\frac{\partial \hat{y}}{\partial z_i} =$

$\frac{\partial \ell}{\partial \hat{y}} =$

β_0 β_1 β_2 z_0 z_1 z_2

$\partial \hat{y} / \partial \beta_i$

$\hat{y} = \sum_{i=0}^{2} \beta_i \cdot z_i$ y

$\partial \ell / \partial \hat{y}$

$\ell = \frac{1}{2}(\hat{y} - y)^2$

Figure 3: Computation graph for the neural network in Figure 2

$\partial \ell / \partial \beta_i = \boxed{\partial \ell / \partial \hat{y}} \cdot \partial \hat{y} / \partial \beta_i$

Hint: $\frac{\partial ReLU(x)}{\partial x} = 1$ if $x > 0$, 0 if $x <= 0$

$$\frac{\partial \ell}{\partial \beta_i} = \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \beta_i}$$

$$\frac{\partial \ell}{\partial \beta_1} =$$

$\partial \ell / \partial \hat{y} = \partial / \partial \hat{y} \left(\frac{1}{2}(\hat{y} - y)^2 \right)$

$$\partial y = \hat{y} - y$$

$$\partial \ell / \partial \beta_i = \partial \ell / \partial \hat{y} \cdot \partial \hat{y} / \partial \beta_i$$

$$= \partial \ell / \partial \hat{y} \cdot \partial / \partial \beta_i \left(\sum_{j=0}^{2} \beta_j \cdot z_j \right)$$

$$= \partial \ell / \partial \hat{y} \cdot z_i$$

$$\frac{\partial \ell}{\partial \alpha_{i,j}} = \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_i} \frac{\partial z_i}{\partial a_i} \frac{\partial a_i}{\partial \alpha_{i,j}} \qquad = (\hat{y} - y) \cdot z_i \qquad \frac{\partial \ell}{\partial \alpha_{1,1}} =$$

$$\frac{\partial \ell}{\partial \hat{y}} = \partial / \partial \hat{y} \left(\tfrac{1}{2} (\hat{y} - y)^2 \right)$$

$$= \hat{y} - y$$

$$\frac{\partial \ell}{\partial z_i} = \partial \ell / \partial \hat{y} \cdot \partial / \partial z_i \left(\sum_{j=0}^{2} \beta_j \cdot z_j \right)$$

$$= \partial \ell / \partial \hat{y} \cdot \beta_i$$

$$\partial \ell / \partial a_i = \partial \ell / \partial z_i \cdot \partial z_i / \partial a_i$$

$$= \partial \ell / \partial z_i \cdot \partial / \partial a_i (\text{ReLU}(a_i))$$

$$\frac{\partial \ell}{\partial \alpha_{ij}} = \partial \ell / \partial a_i \cdot \partial a_i / \partial \alpha_{(ij)}$$

$$= \frac{\partial \ell}{\partial a_i} \cdot \partial / \partial \alpha_{(ij)} \left(\sum_{k=0}^{2} \alpha_{i,k} \cdot x_k \right)$$

$$= \frac{\partial \ell}{\partial a_i} \cdot x_j$$

$$\frac{\partial \ell}{\partial \alpha_{(ij)}} = (\hat{y} - y)(\beta_i)(\partial / \partial a_i \text{ReLU}(a_i)) \cdot (x_j)$$

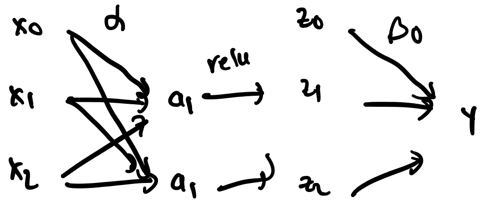
3. **Vector Form:** What are the values of $\frac{\partial \ell}{\partial \beta}$, $\frac{\partial \ell}{\partial \alpha}$?

$$\partial \ell / \partial \alpha_{(ij)} = \frac{\partial \ell}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_i} \cdot \frac{\partial z_i}{\partial a_i} \cdot \frac{\partial a_i}{\partial \alpha_{(ij)}}$$

$$\partial \ell / \partial \beta_i = \underbrace{(\hat{y} - y)} \cdot z_i$$

$$\partial \ell / \partial \vec{\beta} = (\hat{y} - y) \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$$

$$= (\hat{y} - y) \vec{z}$$



$$z_i = \text{ReLU}(a_i)$$

$\hat{\beta} = \beta$ w/o the bias term

$$\partial \ell / \partial \hat{y} = (\hat{y} - y) \leftarrow \text{scalar}$$

$$\partial \hat{y} / \partial z_i = \partial / \partial z_i \left(\sum_{j=0}^{2} \beta_i \cdot z_j \right)$$

$$= \beta_i$$

$$\overset{1\times 1}{\partial \hat{y}} / \partial \vec{z} = \hat{\beta} \leftarrow 2\times 1$$

$$\overset{2\times 1}{\partial z_i} / \underset{2\times 1}{\partial a_j} = \text{if } i \neq j, \ 0$$

$$\text{else, } \partial / \partial a_i \text{ReLU}(a_i)$$

$$= \begin{pmatrix} \partial / \partial a_1 \text{ReLU}(a_1) & 0 \\ 0 & \partial / \partial a_2 \text{ReLU}(a_2) \end{pmatrix}$$

$$\overset{1\times 1}{\partial \ell} / \underset{1\times 1}{\partial \vec{a}} = (\hat{y} - y) \cdot \hat{\beta} \cdot \begin{pmatrix} \partial / \partial a_1 \text{ReLU}(a_1) & 0 \\ 0 & \partial / \partial a_2 \text{ReLU}(a_2) \end{pmatrix}$$

2×1

$$\begin{tikzcd} 2\times1 \arrow[r, bend right, leftrightarrow] & 1\times2 \end{tikzcd}$$

$$= \underset{\underset{1\times1}{\uparrow}}{(\hat{y}-y)} \underbrace{\underset{\underset{2\times2}{\uparrow}}{\begin{pmatrix} \partial/_{\partial a_1} \mathrm{ReLU}(a_1) & 0 \\ 0 & \partial/_{\partial_2} \mathrm{ReLU}(a_2) \end{pmatrix}}}_{2\times1} \underset{\underset{2\times1}{\uparrow}}{\hat{\beta}}$$

$$\frac{\partial \ell}{\partial \alpha_{ij}} = \frac{\partial \ell}{\partial a_i} \cdot \frac{\partial a_i}{\partial \alpha_{ij}}$$

$$= \frac{\partial \ell}{\partial a_i} \cdot \partial/_{\partial \alpha_{ij}} \left(\sum_{k=0}^{2} \alpha_{ik} \cdot x_k \right)$$

$$= \frac{\partial \ell}{\partial a_i} \cdot x_j \quad \curvearrowleft$$

$$\overset{\overset{1\times1}{\downarrow}}{} \underset{\underset{2\times3}{\nearrow}}{\frac{\partial \ell}{\partial \alpha}} = \begin{pmatrix} \frac{\partial \ell}{\partial a_1} \cdot x_0 & \frac{\partial \ell}{\partial a_1} \cdot x_1 & \frac{\partial \ell}{\partial a_1} \cdot x_2 \\ \frac{\partial \ell}{\partial a_2} \cdot x_0 & \frac{\partial \ell}{\partial a_2} \cdot x_1 & \frac{\partial \ell}{\partial 2} \cdot x_2 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{\partial \ell}{\partial a_1} \\ \frac{\partial \ell}{\partial a_2} \end{pmatrix} \begin{pmatrix} x_0 & x_1 & x_2 \end{pmatrix}$$

$$= \frac{\partial \ell}{\partial \vec{a}} \vec{x}^T$$