法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

关注 小象学院

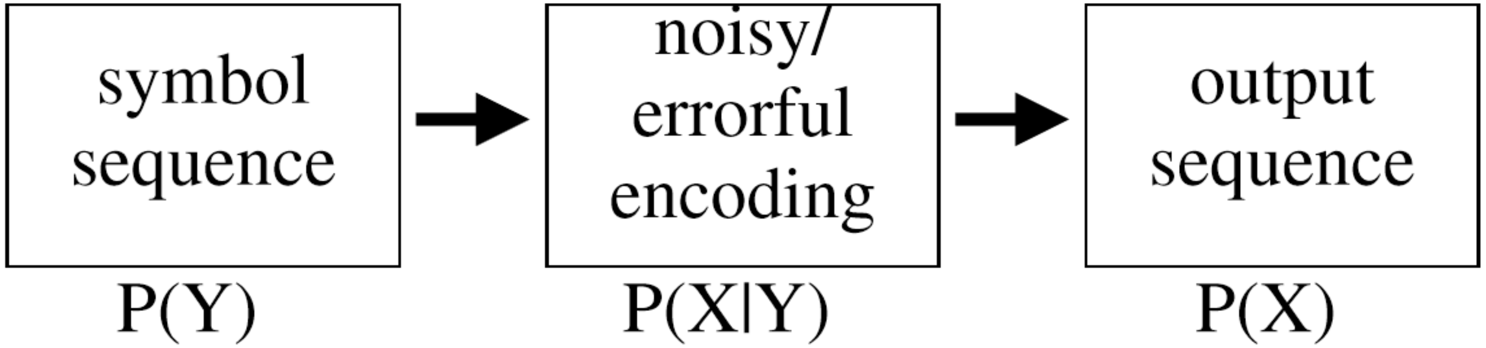
Natural Language Processing – A Machine Learning Approach

Lesson 4: Word

Zengchang Qin (Ph.D.)
http://dsd.future-lab.cn

Word Representation

Noisy Channel Model



Application	Y	X
Speech recognition	spoken words	acoustic signal
Machine translation	words in L_1	words in L_2
Spelling correction	intended words	typed words

How to represent words

Natural language text = sequences of discrete symbols (e.g. words).

Naive representation: one hot vectors in $R^{|V|}$ (very large).

$$\hat{d}_q = \arg\max_d sim(\mathbf{d}, \mathbf{q})$$

Classical IR: document and query vectors are superpositions of word vectors.

Similarly for word classification problems (e.g. document classification).

Issues: sparse, orthogonal representations, semantically weak.

Semantic Similarity

We want richer representations expressing semantic similarity.

Distributional semantics: "**You shall know a word by the company it keeps.**"

— J.R. Firth (1957)

Idea: produce dense vector representations based on the context/use of words.

Three main approaches: count-based, predictive, and task-based.

Count-Based Methods

Define a basis vocabulary C of context words.
Define a word window size w.

Count the basis vocabulary words occurring w words to the left or right of each instance of a target word in the corpus.

Form a vector representation of the target word based on these counts.

Semantic Similarity

... and the cute kitten purred and then ...
... the cute furry cat purred and miaowed ...
... that the small kitten miaowed and she ... .
.. the loud furry dog ran and bit ...

Example basis **vocabulary**: {bit, cute, furry, loud, miaowed, purred, ran, small}.

kitten context words: {cute, purred, small, miaowed}.
cat context words: {cute, furry, miaowed}.
dog context words: {loud, furry, ran, bit}.

Semantic Similarity

... and the cute kitten purred and then ...
... the cute furry cat purred and miaowed ...
... that the small kitten miaowed and she ... .
.. the loud furry dog ran and bit ...

Example basis vocabulary: {bit, cute, furry, loud, miaowed, purred, ran, small}.

$$\mathbf{kitten} = [0, 1, 0, 0, 1, 1, 0, 1]^\top$$
$$\mathbf{cat} = [0, 1, 1, 0, 1, 0, 0, 0]^\top$$
$$\mathbf{dog} = [1, 0, 1, 1, 0, 0, 1, 0]^\top$$

Similarity Calculation

Use inner product or cosine as similarity kernel.

$$sim(\text{kitten}, \text{cat}) = cosine(\mathbf{kitten}, \mathbf{cat}) \approx 0.58$$
$$sim(\text{kitten}, \text{dog}) = cosine(\mathbf{kitten}, \mathbf{dog}) = 0.00$$
$$sim(\text{cat}, \text{dog}) = cosine(\mathbf{cat}, \mathbf{dog}) \approx 0.29$$

Cosine has the advantage that it's a norm-invariant metric.

$$cosine(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{u}|| \times ||\mathbf{v}||}$$

TF-IDF

Not all features are equal: we must distinguish counts that are high because the are informative from those that are just independently frequent contexts.

Many normalization methods: TF-IDF, PMI, etc.

One-Hot Representation

Learning count based vectors produces an embedding matrix in $R^{|V| \times |C|}$

$$\mathbf{E} = \begin{array}{c} \\ \text{kitten} \\ \text{cat} \\ \text{dog} \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccc} \text{bit} & \text{cute} & \text{furry} & \ldots \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & \ldots \\ 0 & 1 & 1 & \ldots \\ 1 & 0 & 1 & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \end{array} \qquad onehot_{cat} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \mathbf{cat} = onehot_{cat}^{\top} \mathbf{E}$$

Rows are word vectors, so we can retrieve them with **one hot** vectors.

Embedding

The generic idea behind embedding learning:

1. Collect instances ti ∈ inst(t) of a word t of vocabulary V.
2. For each instance, collect its context words c(ti) (e.g. k-word window).
3. Define some score function score(ti , c(ti); θ, E)
4. Define a loss: $$L = -\sum_{t \in V} \sum_{t_i \in inst(t)} score(t_i, c(t_i); \theta, \mathbf{E})$$
5. Estimate parameters: $$\hat{\theta}, \hat{\mathbf{E}} = \arg\min_{\theta, \mathbf{E}} L$$
6. Use E as the embedding matrix.

Neural Embedding Models: C&W (Collobert et al. 2011)

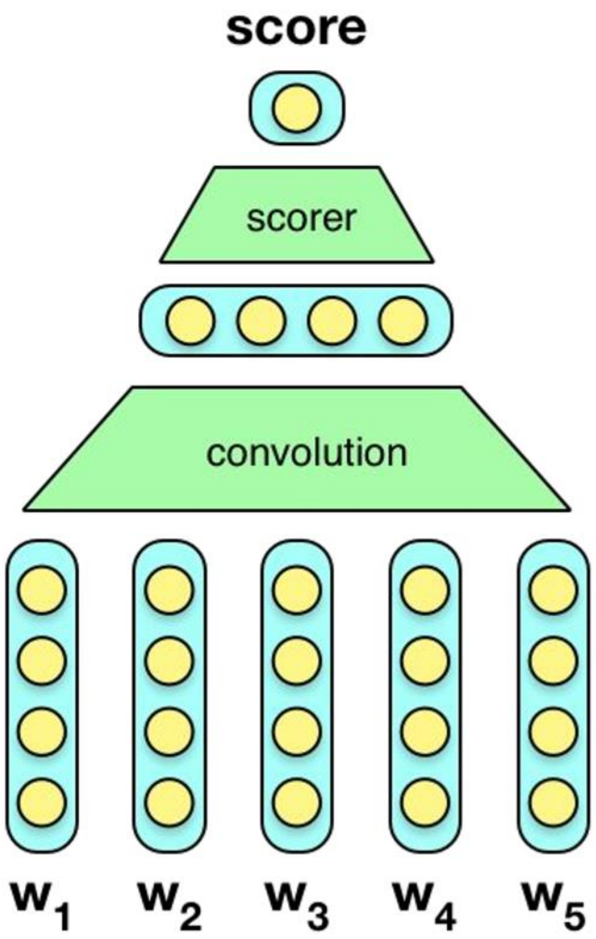
Embed all words in a sentence with E.

Shallow convolution over embeddings.

MLP projects output of convolution to a scalar score. Convolutions and MLP are parameterised by a set of weights θ.

Overall network models a function over sentences s:

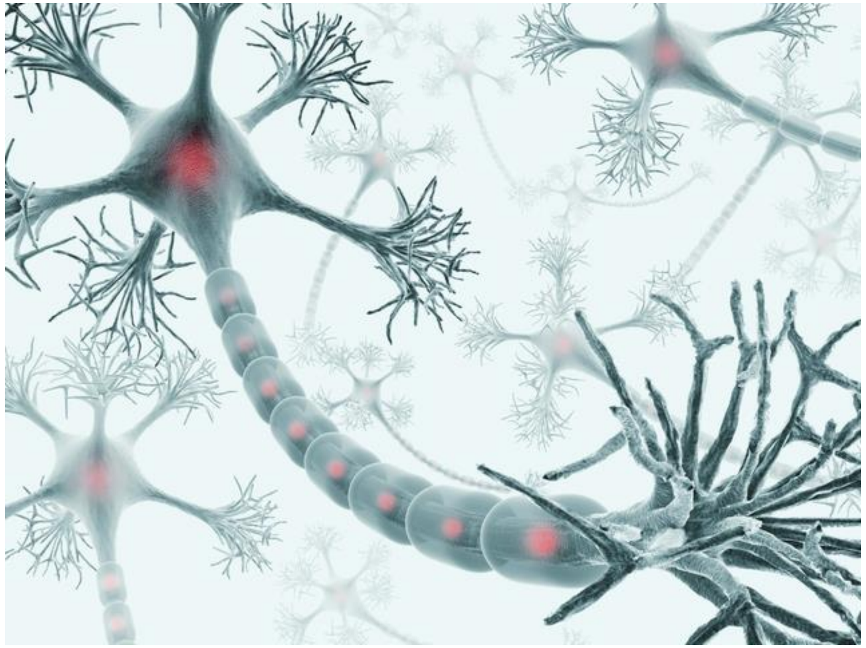
$$g_{\theta,E}(s) = f_{\theta}(embed_{E}(s))$$

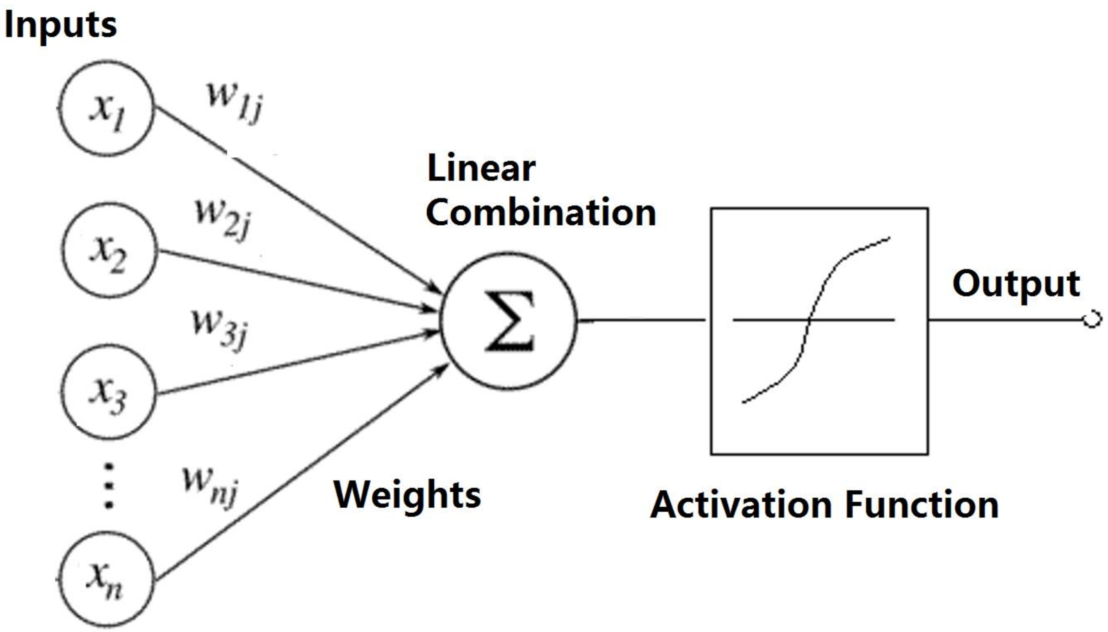


Basics of Neural Network

The Neuron

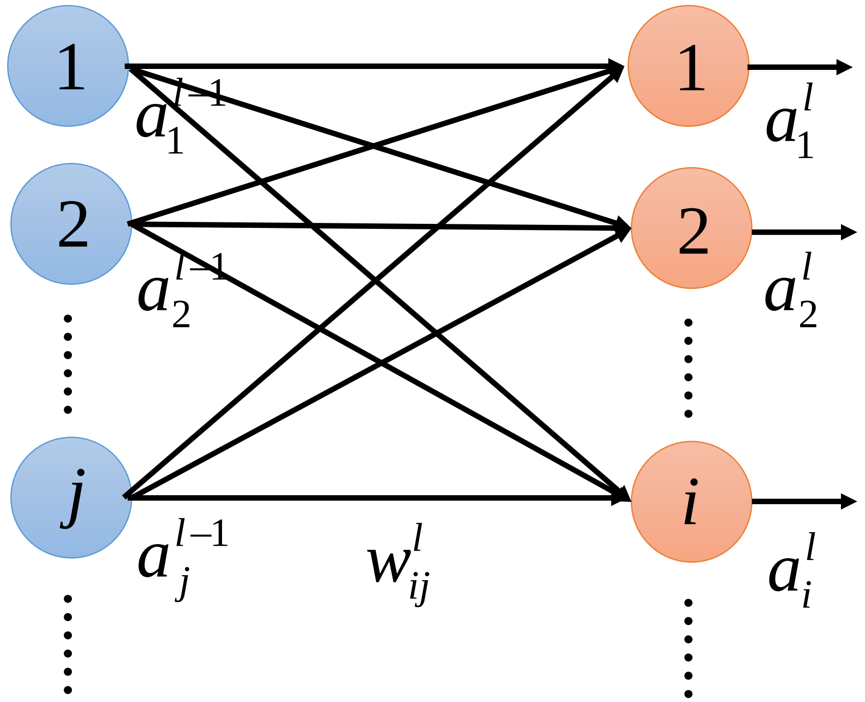
Neural Networks (Artificial Neural Networks, Precisely) are always with story of its biological counterpart, however, with the advancement of A.I., we now seriously believe it is a mathematical model of "imitating".





Full Connected Networks

Fully Connected Feed-Forward Neural Networks



Layer $l-1$

N_{l-1} nodes

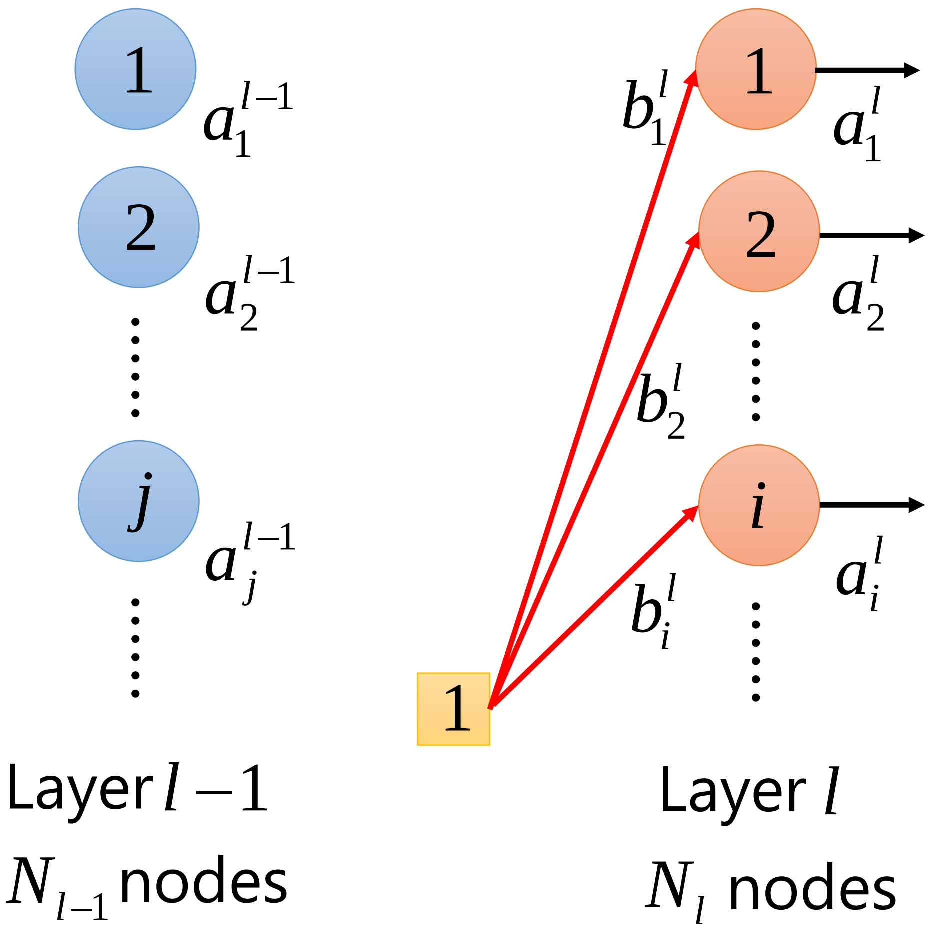
Layer l

N_l nodes

$w_{ij}^l \longrightarrow$ Layer $l-1$ to Layer l

$$W^l = \left.\overbrace{\begin{bmatrix} w_{11}^l & w_{12}^l & \cdots \\ w_{21}^l & w_{22}^l & \\ \vdots & & \ddots \\ & & \end{bmatrix}}^{N_{l-1}}\right\} N_l$$

Bias

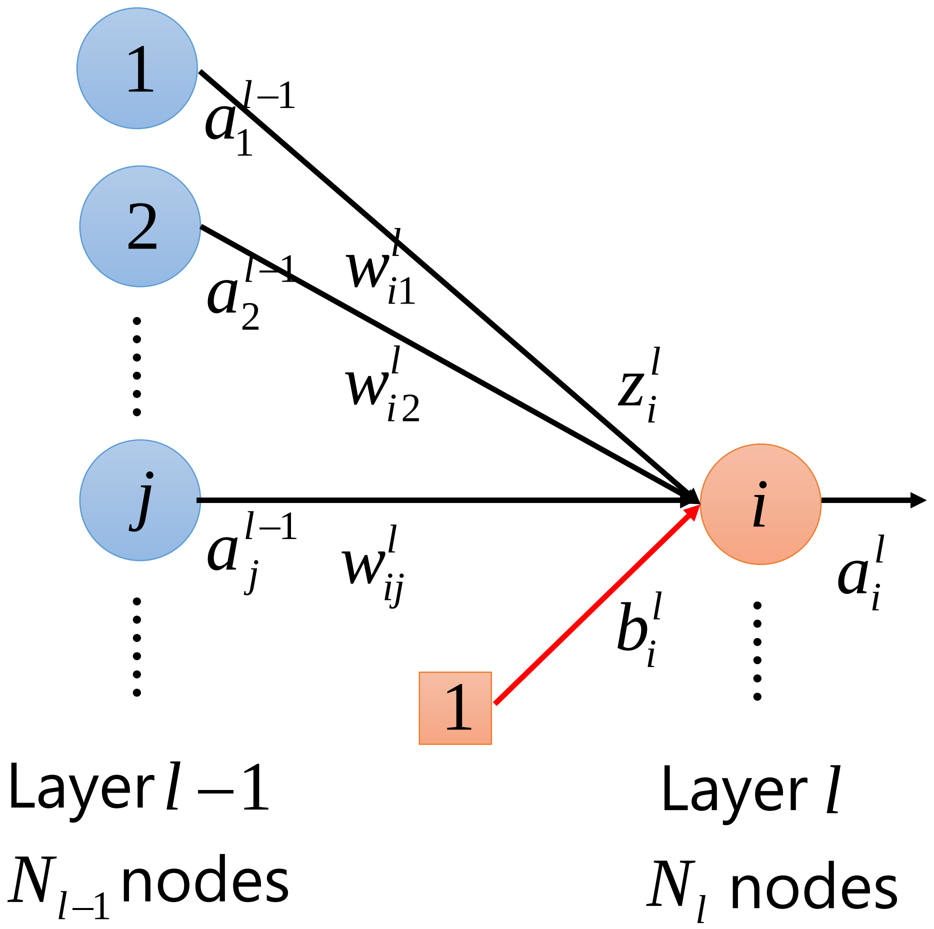


b_i^l : bias for neuron i at layer l

$$b^l = \begin{bmatrix} b_1^l \\ b_2^l \\ \vdots \\ b_i^l \\ \vdots \end{bmatrix}$$

bias for all neurons in layer l

Linear Combination



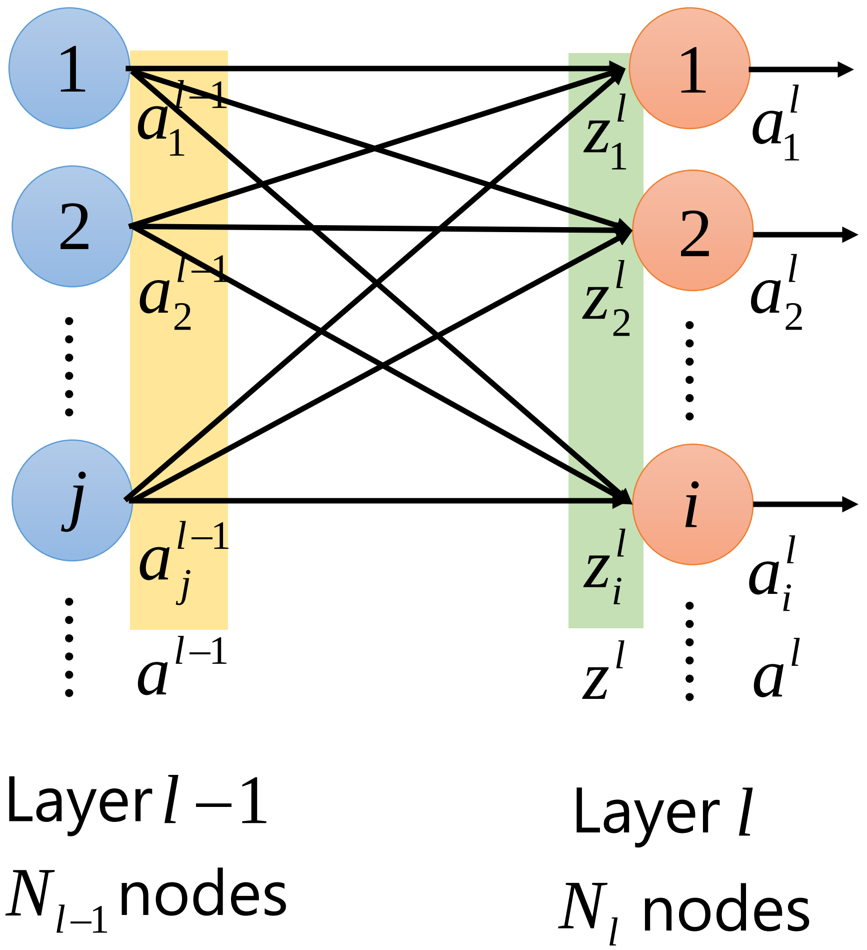
z_i^l : input of the activation function for neuron i at layer l

z^l : input of the activation function all the neurons in layer l

$$z_i^l = w_{i1}^l a_1^{l-1} + w_{i2}^l a_2^{l-1} \ldots + b_i^l$$

$$z_i^l = \sum_{j=1}^{N_{l-1}} w_{ij}^l a_j^{l-1} + b_i^l$$

Inputs-Outputs Relations



$$z_1^l = w_{11}^l a_1^{l-1} + w_{12}^l a_2^{l-1} + \cdots + b_1^l$$

$$z_2^l = w_{21}^l a_1^{l-1} + w_{22}^l a_2^{l-1} + \cdots + b_2^l$$

$$\vdots$$

$$z_i^l = w_{i1}^l a_1^{l-1} + w_{i2}^l a_2^{l-1} + \cdots + b_i^l$$

$$\vdots$$

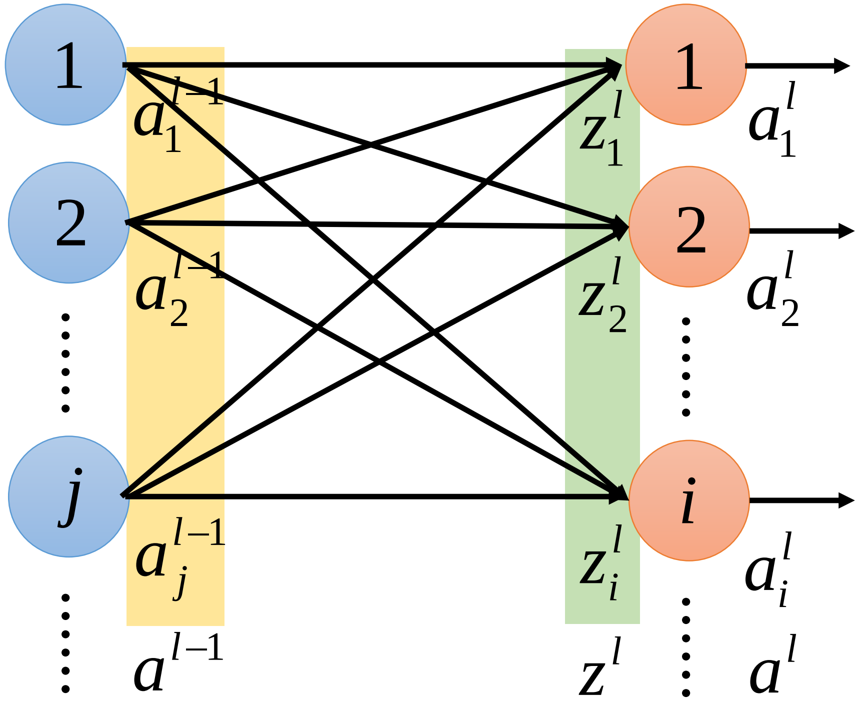
$$\Downarrow$$

$$\begin{bmatrix} z_1^l \\ z_2^l \\ \vdots \\ z_i^l \\ \vdots \end{bmatrix} = \begin{bmatrix} w_{11}^l & w_{12}^l & \cdots \\ w_{21}^l & w_{22}^l & \\ \vdots & & \ddots \end{bmatrix} \begin{bmatrix} a_1^{l-1} \\ a_2^{l-1} \\ \vdots \\ a_i^{l-1} \\ \vdots \end{bmatrix} + \begin{bmatrix} b_1^l \\ b_2^l \\ \vdots \\ b_i^l \\ \vdots \end{bmatrix}$$

$$\Downarrow$$

$$z^l = W^l a^{l-1} + b^l$$

Inputs-Outputs Relations (Activation)



Layer $l-1$

N_{l-1} nodes

Layer l

N_l nodes

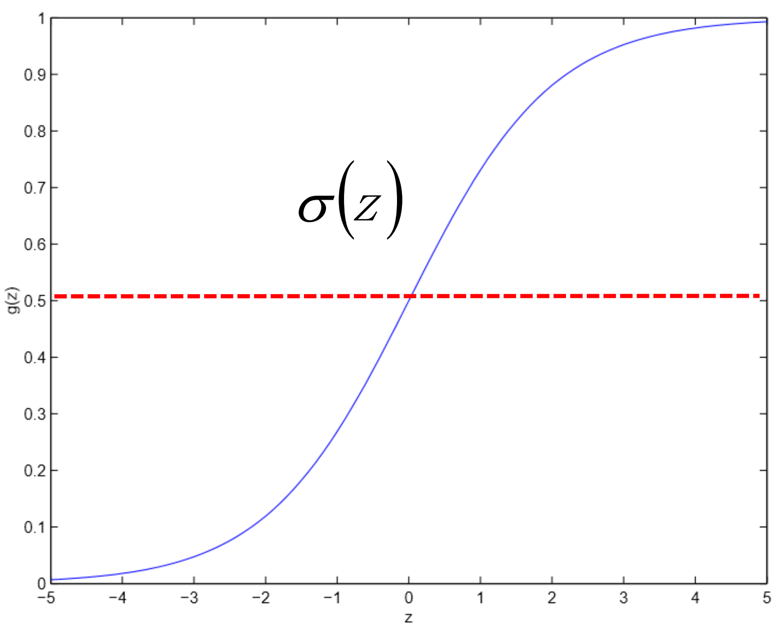
$$a_i^l = \sigma\left(z_i^l\right)$$

$$\begin{bmatrix} a_1^l \\ a_2^l \\ \vdots \\ a_i^l \\ \vdots \end{bmatrix} = \begin{bmatrix} \sigma\left(z_1^l\right) \\ \sigma\left(z_2^l\right) \\ \vdots \\ \sigma\left(z_i^l\right) \\ \vdots \end{bmatrix}$$

$$a^l = \sigma\left(z^l\right)$$

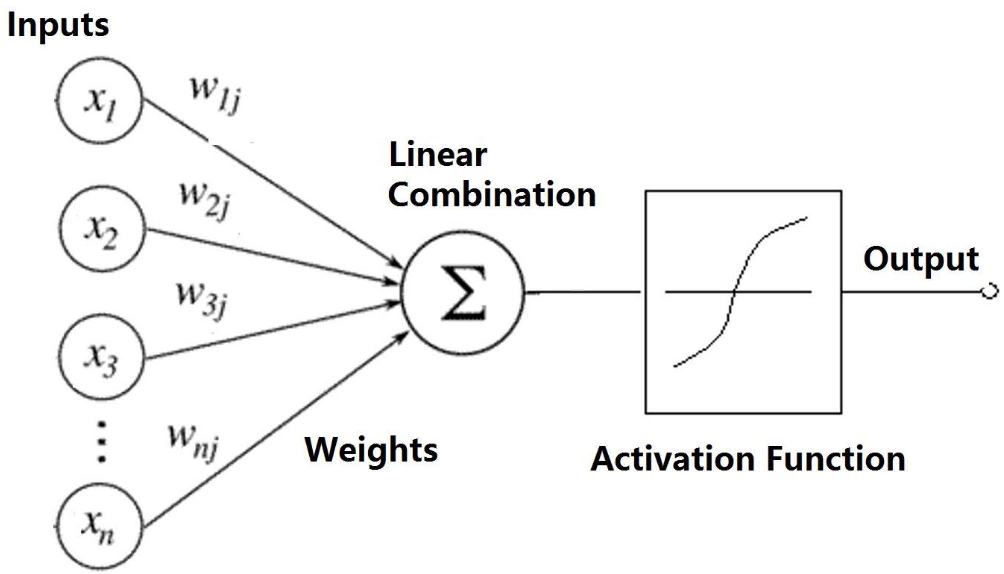
$$a^l = \sigma\left(W^l a^{l-1} + b^l\right)$$

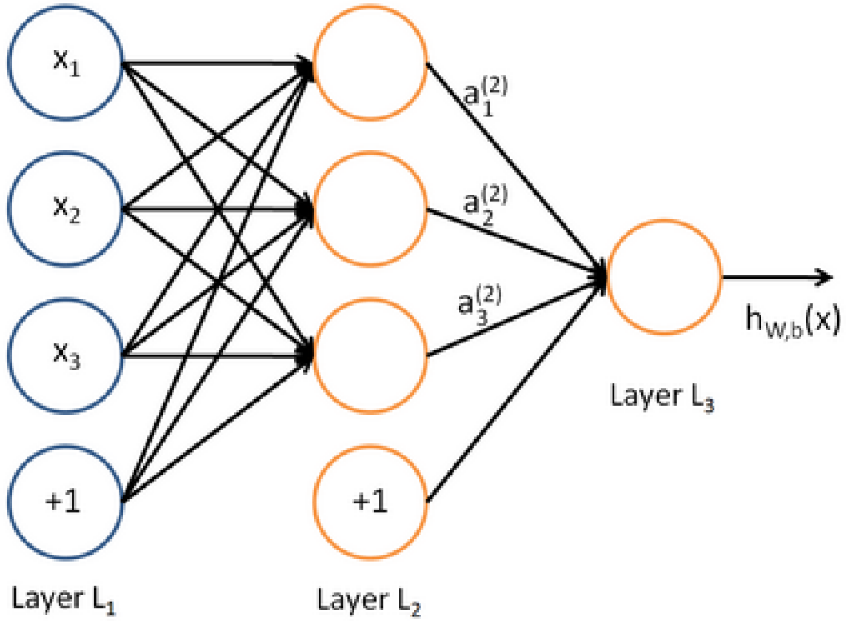
Sigmoid Function



NN and LR

A neural network = running several logistic regressions at the same time.

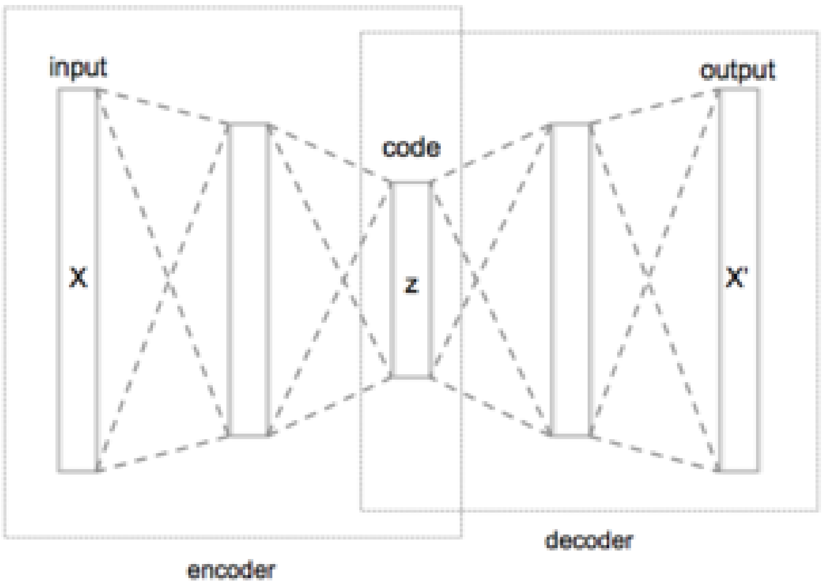


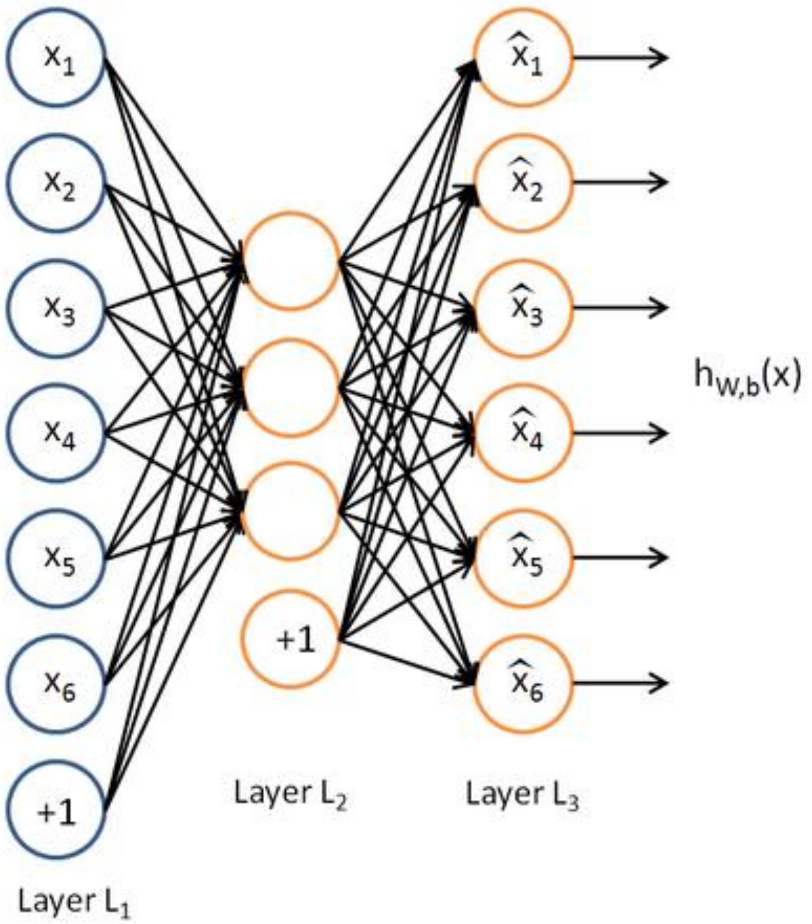


The LRs are then feed into another logistic regression function.

Auto-Encoder

Auto-Encoder is a neural network used to learn efficient coding. Architecturally, the simplest form of an autoencoder is a feedforward, non-recurrent neural network





Back Propagation

A Simple Example of BP

Backpropagation: a simple example

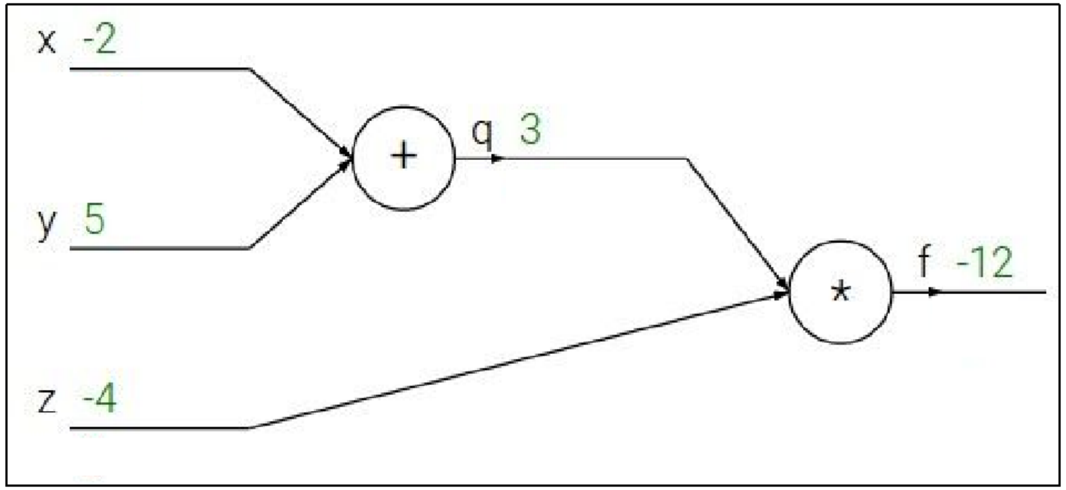
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



A Simple Example of BP

Backpropagation: a simple example

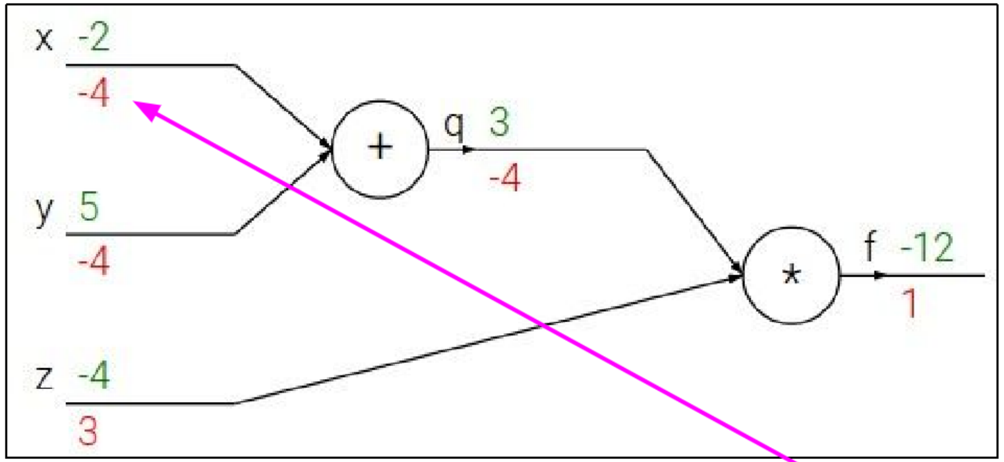
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

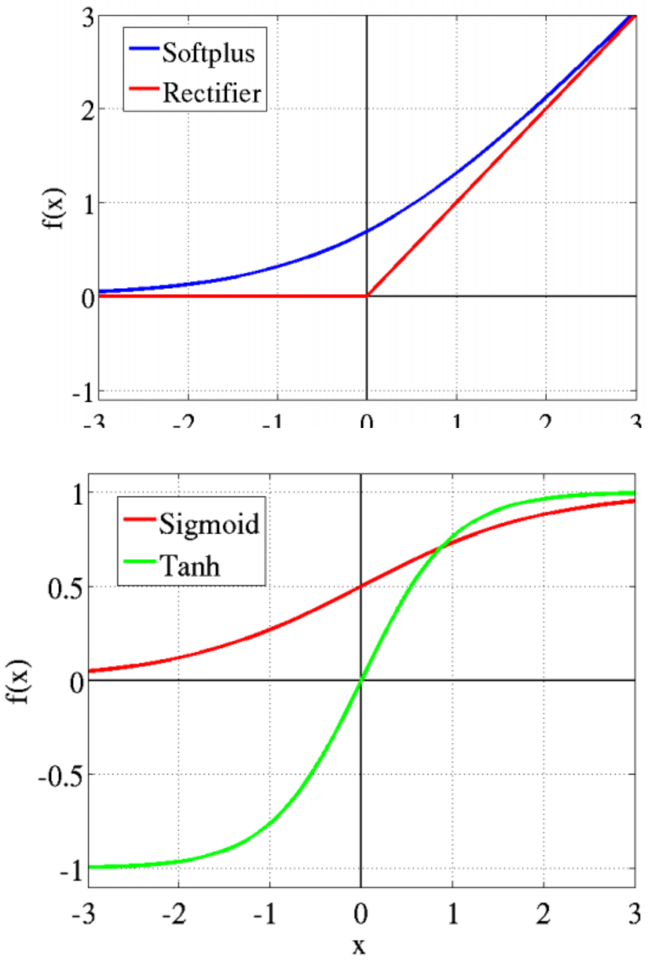


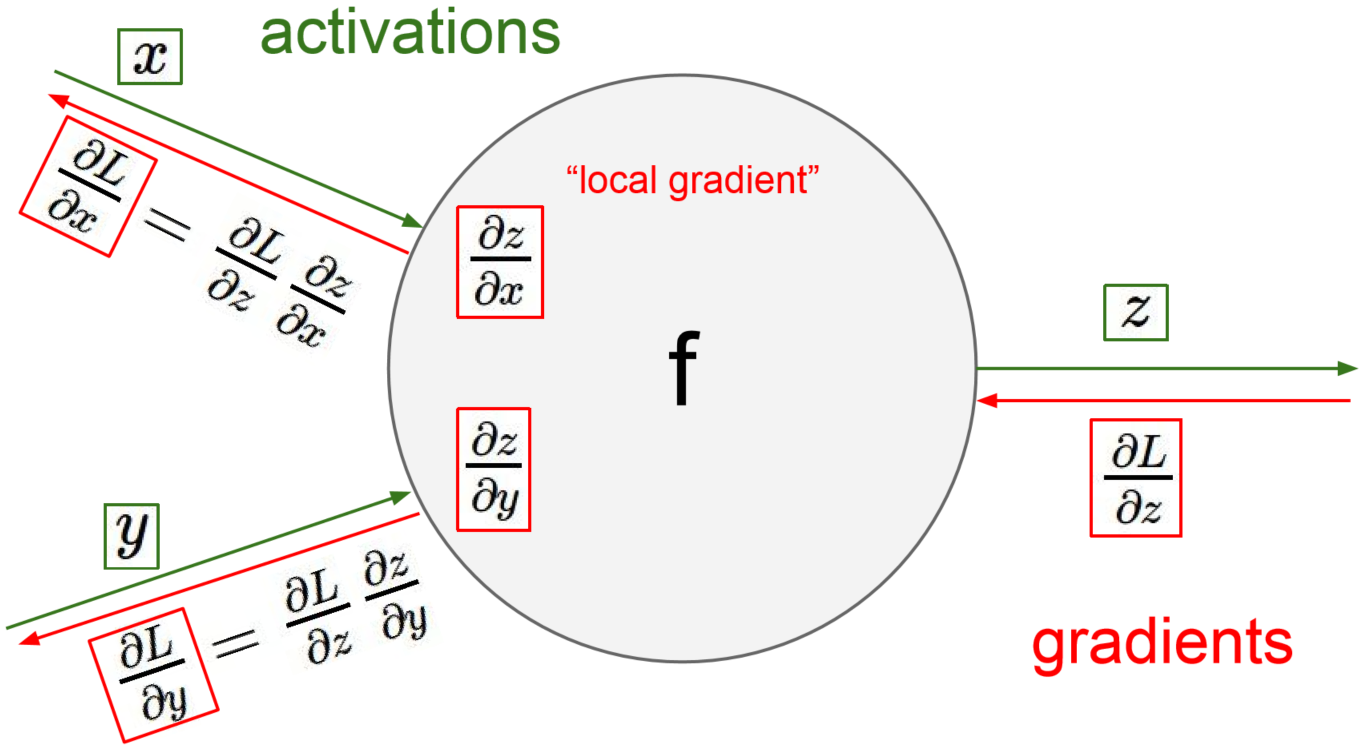
$$\frac{\partial f}{\partial x}$$

Chain rule:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}$$

Illustration of Back-Propagation

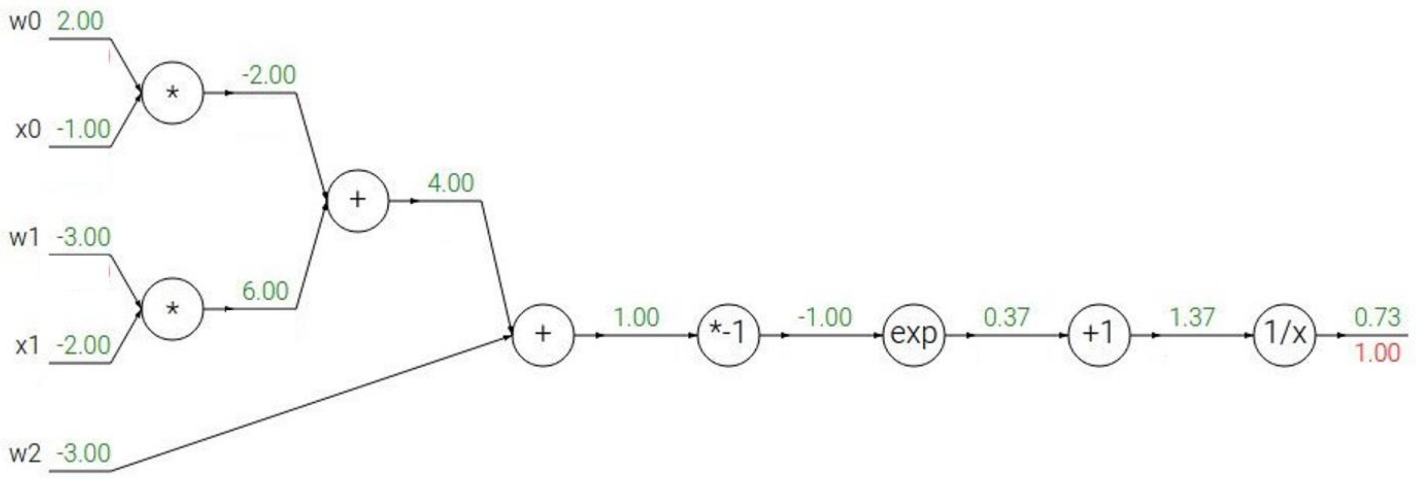




Back Propagation through layers.

A Simple Example of BP

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

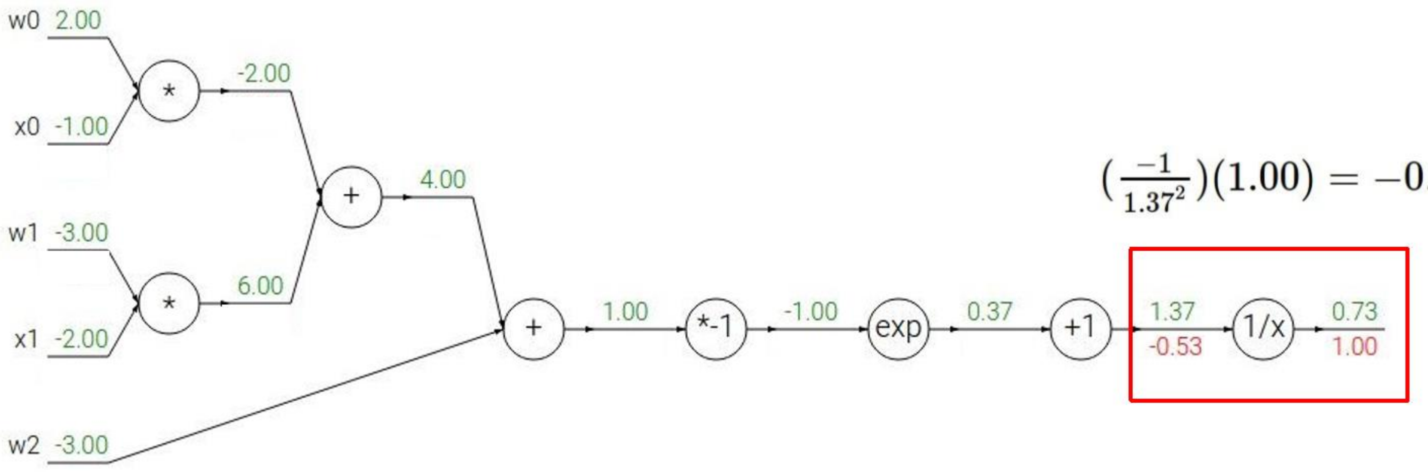
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

A Simple Example of BP

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$\left(\frac{-1}{1.37^2}\right)(1.00) = -0.53$$

$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

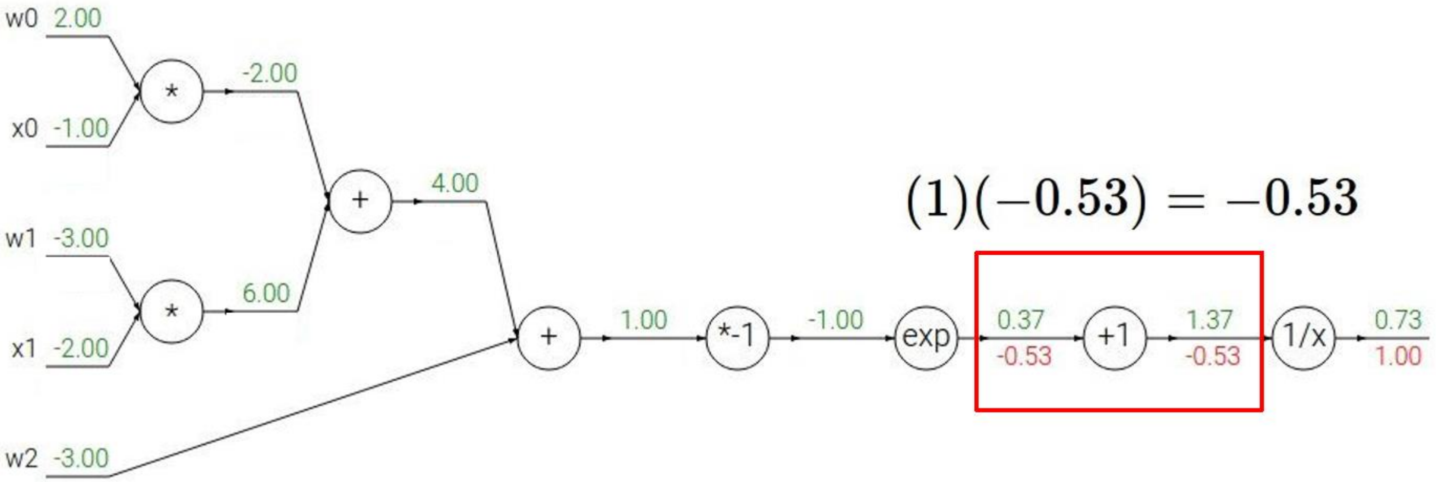
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$\boxed{f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2}$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

A Simple Example of BP

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

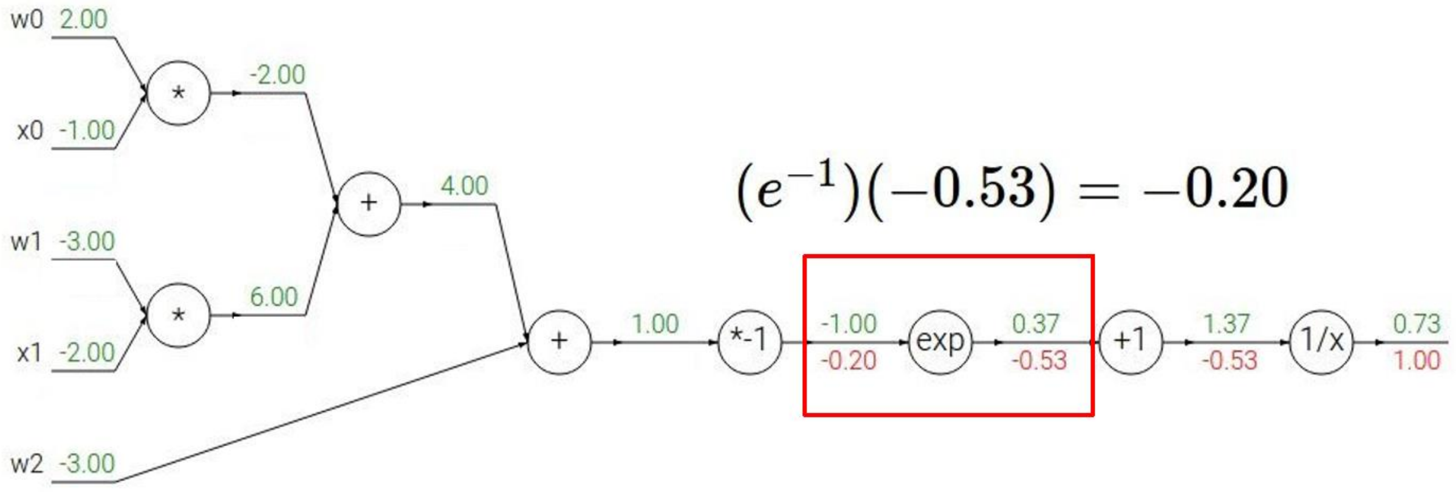
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$\boxed{f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1}$$

A Simple Example of BP

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$(e^{-1})(-0.53) = -0.20$$

$$\boxed{f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x}$$

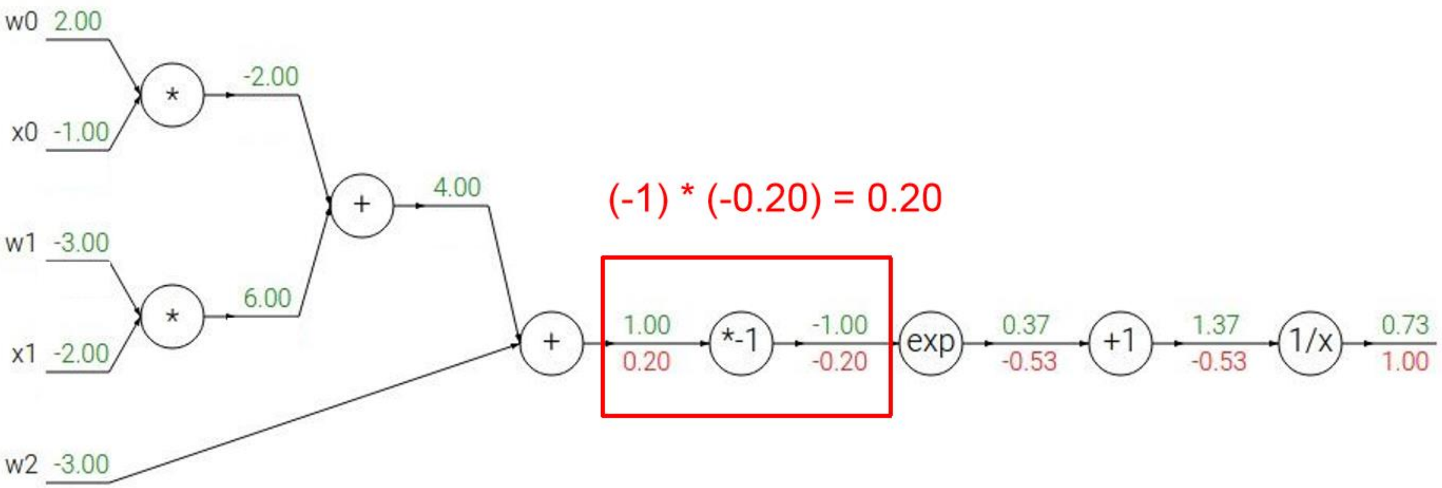
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

A Simple Example of BP

Another example: $f(w,x) = \dfrac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$



$f(x) = e^x \quad \rightarrow \quad \dfrac{df}{dx} = e^x$

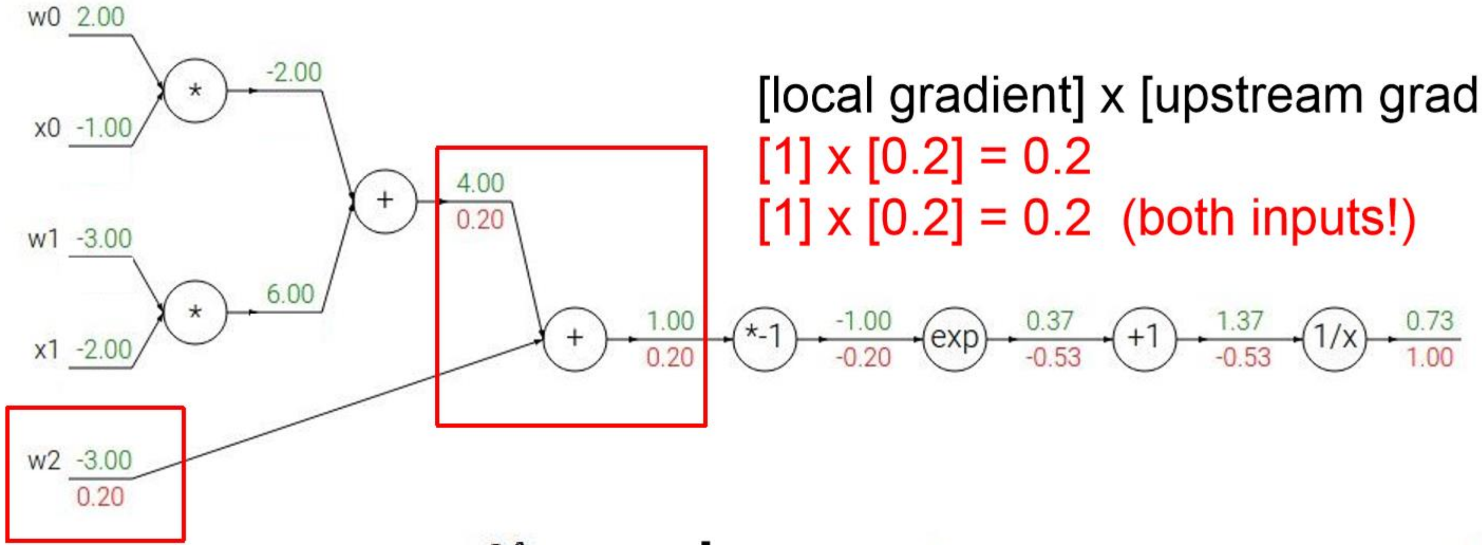
$f_a(x) = ax \quad \rightarrow \quad \dfrac{df}{dx} = a$

$f(x) = \dfrac{1}{x} \quad \rightarrow \quad \dfrac{df}{dx} = -1/x^2$

$f_c(x) = c + x \quad \rightarrow \quad \dfrac{df}{dx} = 1$

A Simple Example of BP

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



[local gradient] x [upstream gradient]

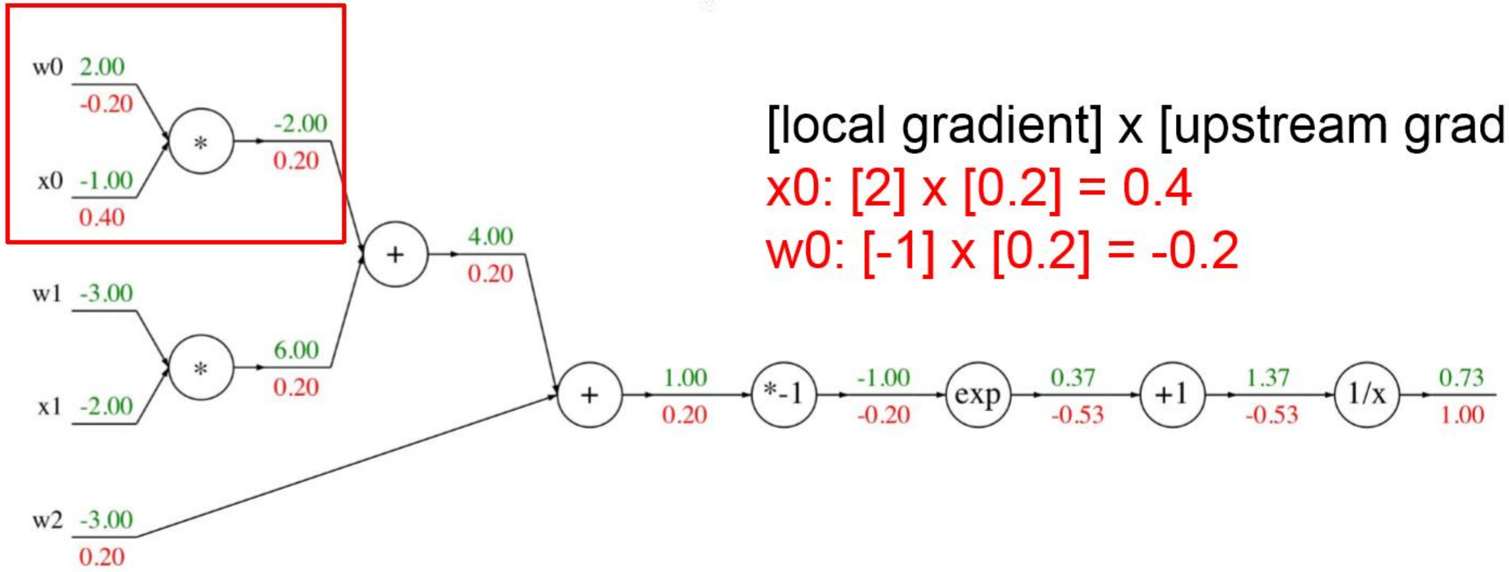
[1] x [0.2] = 0.2

[1] x [0.2] = 0.2 (both inputs!)

$f(x) = e^x$	$\rightarrow$	$\frac{df}{dx} = e^x$	$f(x) = \frac{1}{x}$	$\rightarrow$	$\frac{df}{dx} = -1/x^2$
$f_a(x) = ax$	$\rightarrow$	$\frac{df}{dx} = a$	$f_c(x) = c + x$	$\rightarrow$	$\frac{df}{dx} = 1$

A Simple Example of BP

Another example: $f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$



[local gradient] x [upstream gradient]

x0: [2] x [0.2] = 0.4

w0: [-1] x [0.2] = -0.2

$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$

$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$

$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$

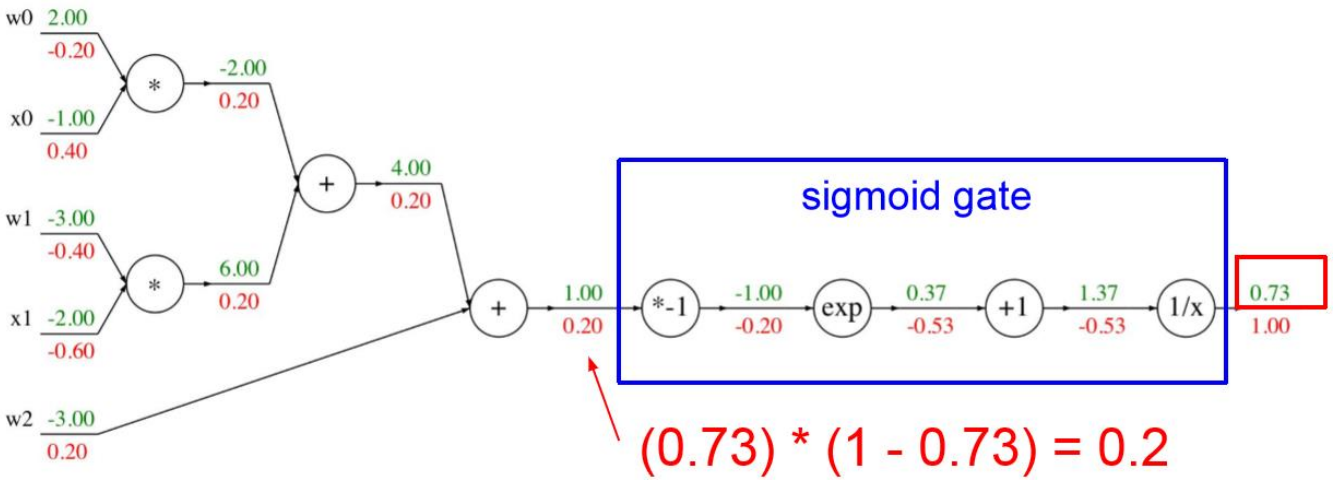
$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$

A Simple Example of BP

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

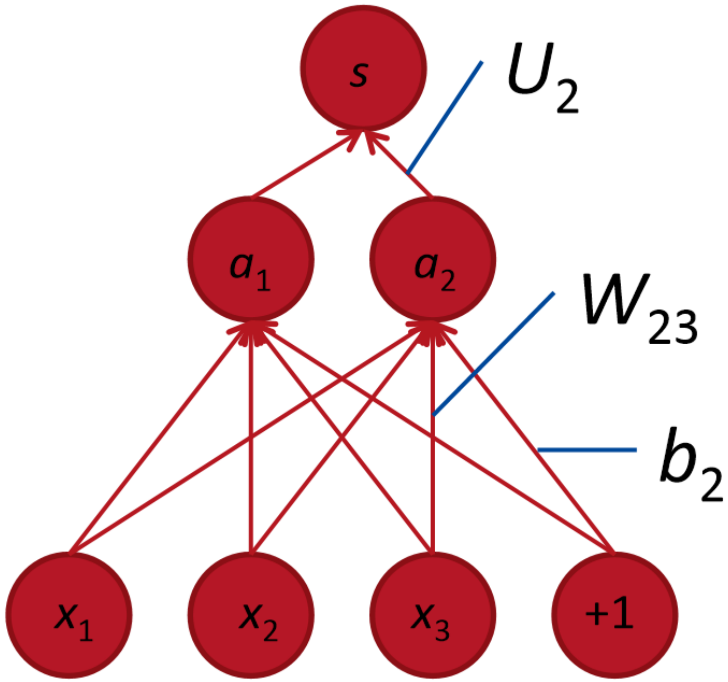
$$\boxed{\sigma(x) = \frac{1}{1+e^{-x}}}$$ sigmoid function

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$



Back-Propagation

W_{23} is only used to compute a_2, the error is back-propagated through a_2.



$$a_i = f(z_i) \quad z_i = W_{i\cdot}x + b_i = \sum_{j=1}^{3} W_{ij}x_j + b_i$$

$$\frac{\partial s}{\partial W} = \frac{\partial}{\partial W}U^T a = \frac{\partial}{\partial W}U^T f(z) = \frac{\partial}{\partial W}U^T f(Wx+b)$$

$$\frac{\partial}{\partial W_{ij}}U^T a \rightarrow \frac{\partial}{\partial W_{ij}}U_i a_i$$

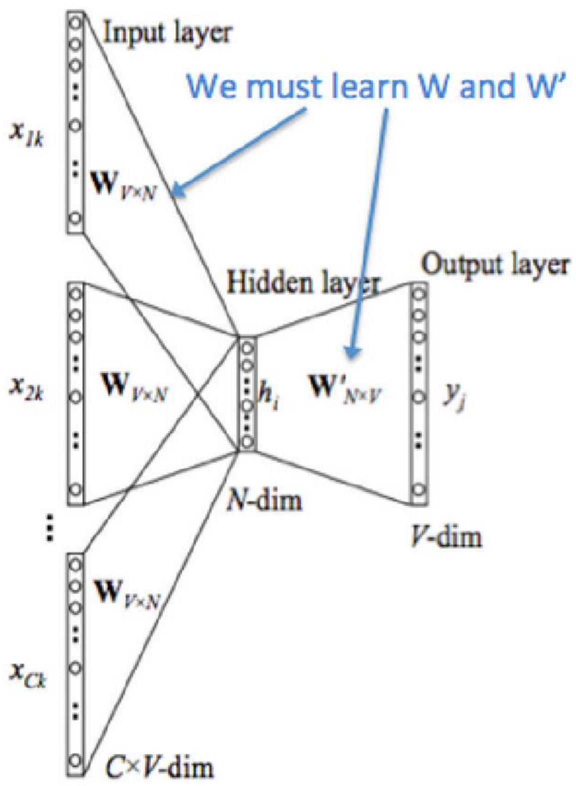
$$\begin{aligned}
U_i \frac{\partial}{\partial W_{ij}} a_i &= U_i \frac{\partial a_i}{\partial z_i}\frac{\partial z_i}{\partial W_{ij}} = U_i \frac{\partial f(z_i)}{\partial z_i}\frac{\partial z_i}{\partial W_{ij}} \\
&= U_i f'(z_i)\frac{\partial z_i}{\partial W_{ij}} = U_i f'(z_i)\frac{\partial W_{i\cdot}x + b_i}{\partial W_{ij}} \\
&= U_i f'(z_i)\frac{\partial}{\partial W_{ij}}\sum_k W_{ik}x_k = \underbrace{U_i f'(z_i)}\, x_j \\
&= \delta_i \quad x_j
\end{aligned}$$

Local error signal

Local input signal

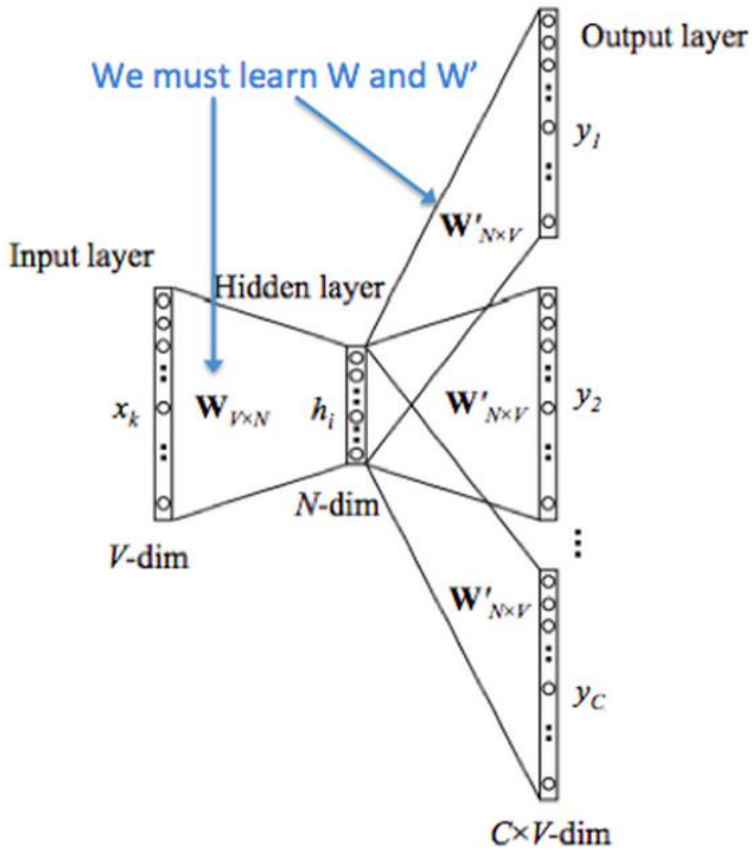
Neural Language Model - CBOW

1. We generate our one hot word vectors for the input context of size $m : (x^{(c-m)}, \ldots, x^{(c-1)}, x^{(c+1)}, \ldots, x^{(c+m)} \in \mathbb{R}^{|V|})$.
2. We get our embedded word vectors for the context $(v_{c-m} = \mathcal{V}x^{(c-m)}, v_{c-m+1} = \mathcal{V}x^{(c-m+1)}, \ldots, v_{c+m} = \mathcal{V}x^{(c+m)} \in \mathbb{R}^n)$
3. Average these vectors to get $\hat{v} = \frac{v_{c-m}+v_{c-m+1}+\ldots+v_{c+m}}{2m} \in \mathbb{R}^n$
4. Generate a score vector $z = \mathcal{U}\hat{v} \in \mathbb{R}^{|V|}$. As the dot product of similar vectors is higher, it will push similar words close to each other in order to achieve a high score.
5. Turn the scores into probabilities $\hat{y} = \text{softmax}(z) \in \mathbb{R}^{|V|}$.
6. We desire our probabilities generated, $\hat{y} \in \mathbb{R}^{|V|}$, to match the true probabilities, $y \in \mathbb{R}^{|V|}$, which also happens to be the one hot vector of the actual word.

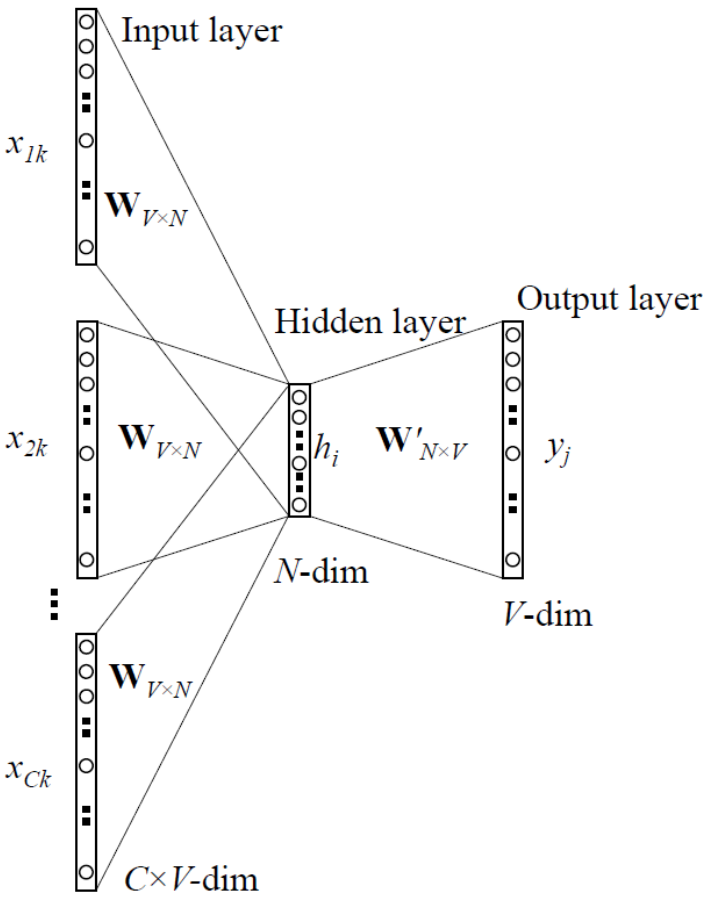


Word2Vec Skip-Gram

1. We generate our one hot input vector $x \in \mathbb{R}^{|V|}$ of the center word.
2. We get our embedded word vector for the center word $v_c = \mathcal{V}x \in \mathbb{R}^n$
3. Generate a score vector $z = \mathcal{U}v_c$.
4. Turn the score vector into probabilities, $\hat{y} = \text{softmax}(z)$. Note that $\hat{y}_{c-m}, \ldots, \hat{y}_{c-1}, \hat{y}_{c+1}, \ldots, \hat{y}_{c+m}$ are the probabilities of observing each context word.
5. We desire our probability vector generated to match the true probabilities which is $y^{(c-m)}, \ldots, y^{(c-1)}, y^{(c+1)}, \ldots, y^{(c+m)}$, the one hot vectors of the actual output.



Word2Vec



Word2vec is a group of related models that are used to produce word embedding. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words.