Advanced Text Analysis

Day 4 / Session 1

Petro Tolochko

Neural Networks & Transformer Models

Vanilla NN

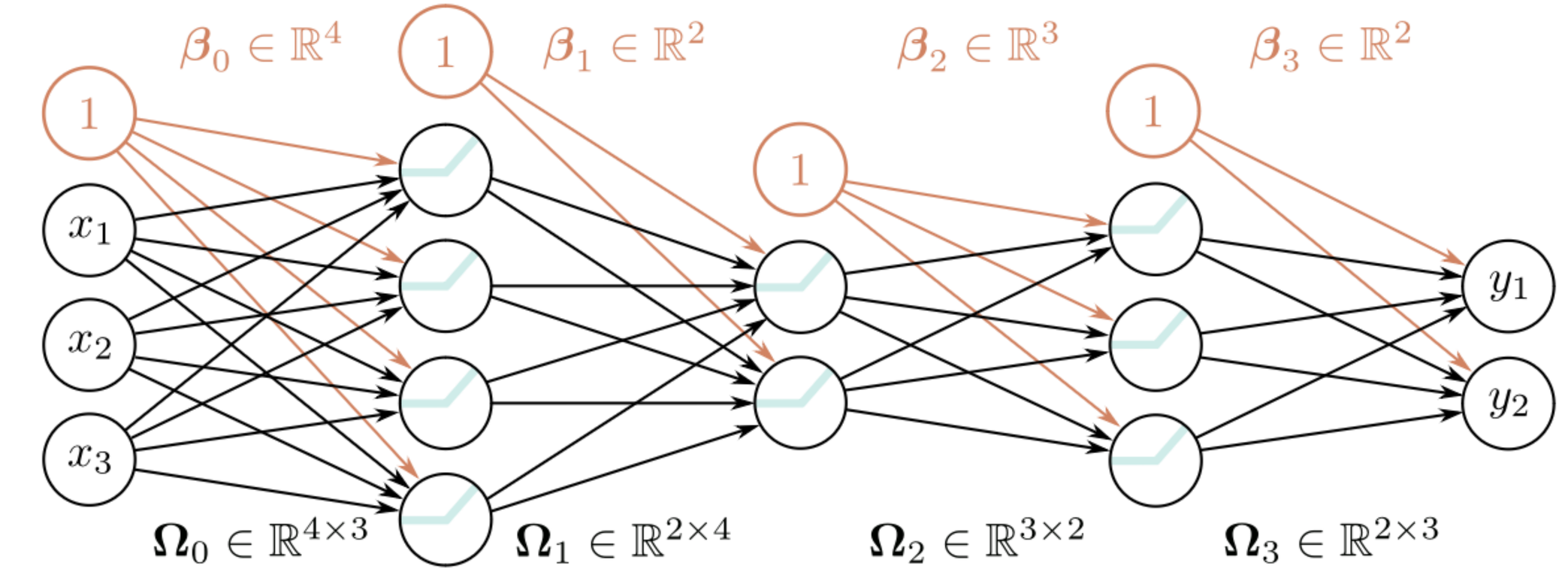
- Neural Network

Vanilla NN

- Neural Network
- Neural?

Vanilla NN

- Neural Network
- Neural?
- Network?



Input, $\mathbf{x}$

$D_i = 3$

Hidden layer, $\mathbf{h}_1$

$D_1 = 4$

Hidden layer, $\mathbf{h}_2$

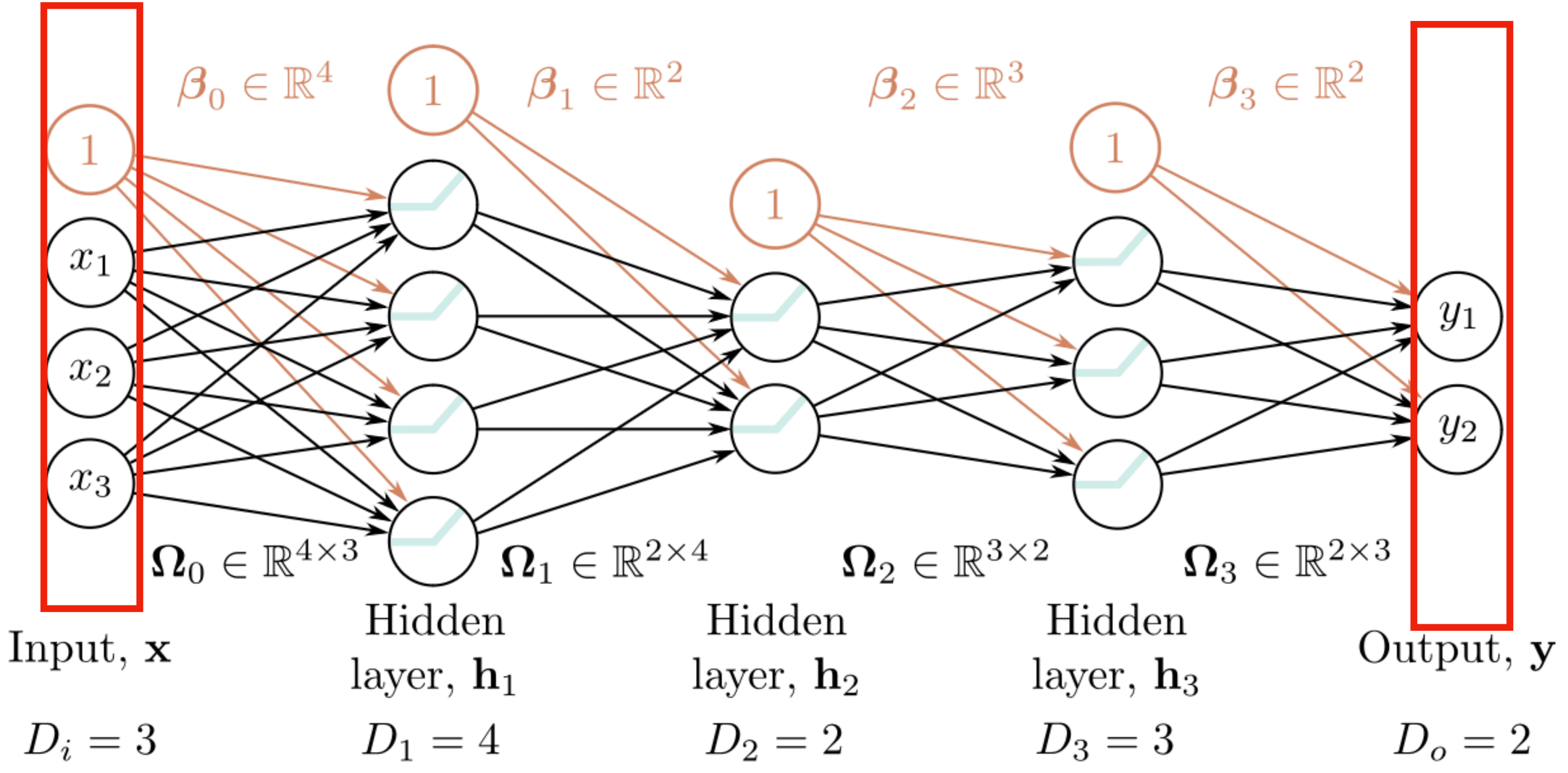
$D_2 = 2$

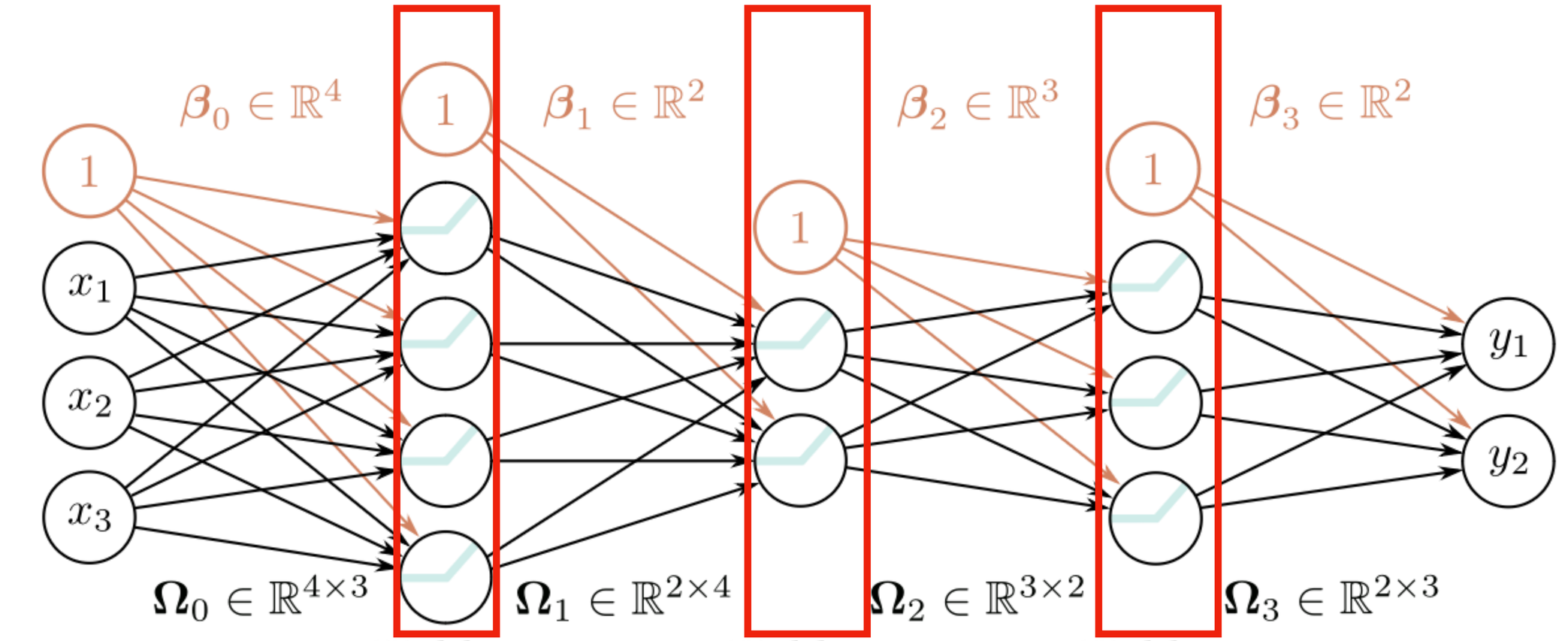
Hidden layer, $\mathbf{h}_3$

$D_3 = 3$

Output, $\mathbf{y}$

$D_o = 2$


$\boldsymbol{\beta}_0 \in \mathbb{R}^4$
$\boldsymbol{\beta}_1 \in \mathbb{R}^2$
$\boldsymbol{\beta}_2 \in \mathbb{R}^3$
$\boldsymbol{\beta}_3 \in \mathbb{R}^2$
1
x_1
x_2
x_3
y_1
y_2
$\boldsymbol{\Omega}_0 \in \mathbb{R}^{4\times3}$
$\boldsymbol{\Omega}_1 \in \mathbb{R}^{2\times4}$
$\boldsymbol{\Omega}_2 \in \mathbb{R}^{3\times2}$
$\boldsymbol{\Omega}_3 \in \mathbb{R}^{2\times3}$
Input, $\mathbf{x}$
Hidden layer, $\mathbf{h}_1$
Hidden layer, $\mathbf{h}_2$
Hidden layer, $\mathbf{h}_3$
Output, $\mathbf{y}$
$D_i = 3$
$D_1 = 4$
$D_2 = 2$
$D_3 = 3$
$D_o = 2$



Input, $\mathbf{x}$

$D_i = 3$

Hidden layer, $\mathbf{h}_1$

$D_1 = 4$

Hidden layer, $\mathbf{h}_2$

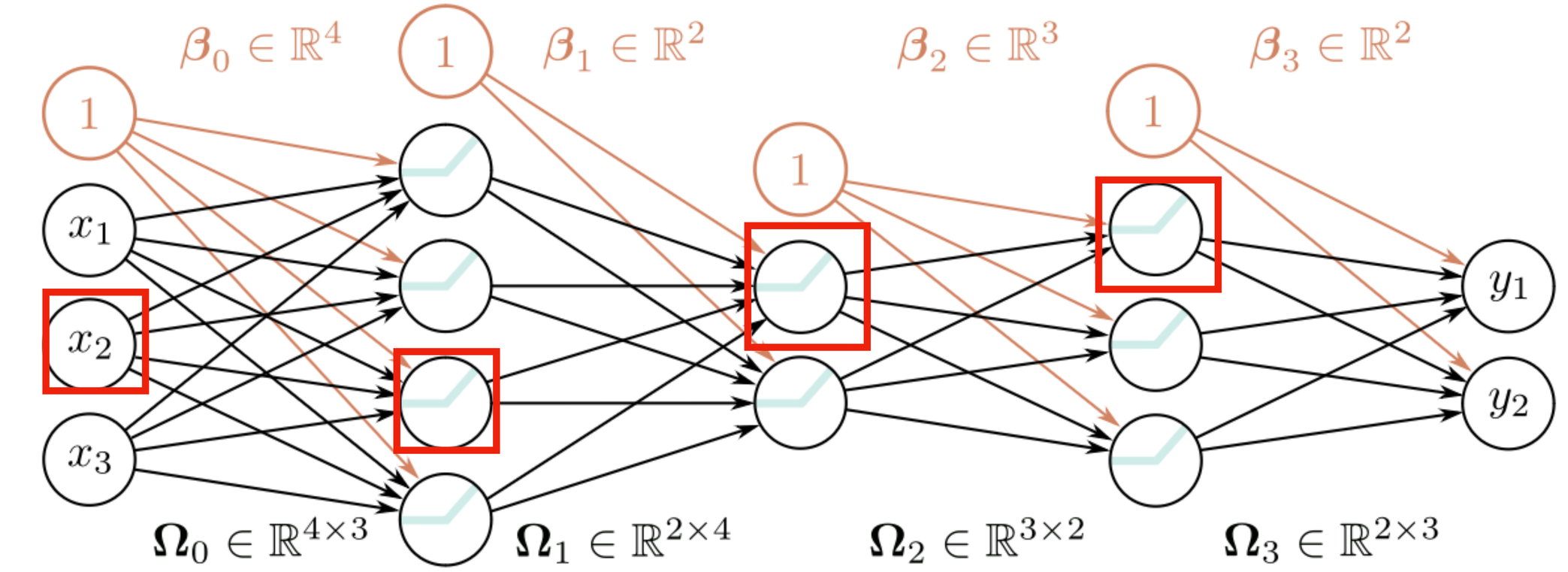
$D_2 = 2$

Hidden layer, $\mathbf{h}_3$

$D_3 = 3$

Output, $\mathbf{y}$

$D_o = 2$



Input, $\mathbf{x}$

$D_i = 3$

Hidden layer, $\mathbf{h}_1$

$D_1 = 4$

Hidden layer, $\mathbf{h}_2$

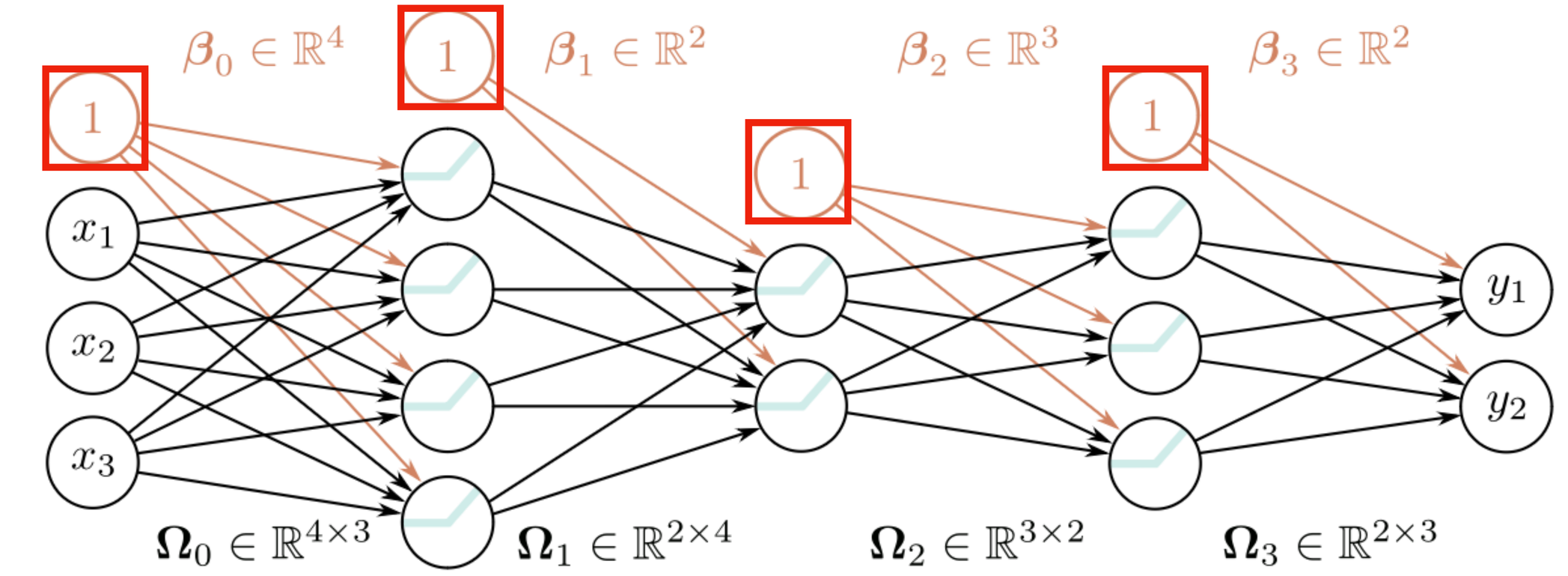
$D_2 = 2$

Hidden layer, $\mathbf{h}_3$

$D_3 = 3$

Output, $\mathbf{y}$

$D_o = 2$



Input, $\mathbf{x}$

$D_i = 3$

Hidden layer, $\mathbf{h}_1$

$D_1 = 4$

Hidden layer, $\mathbf{h}_2$

$D_2 = 2$

Hidden layer, $\mathbf{h}_3$

$D_3 = 3$

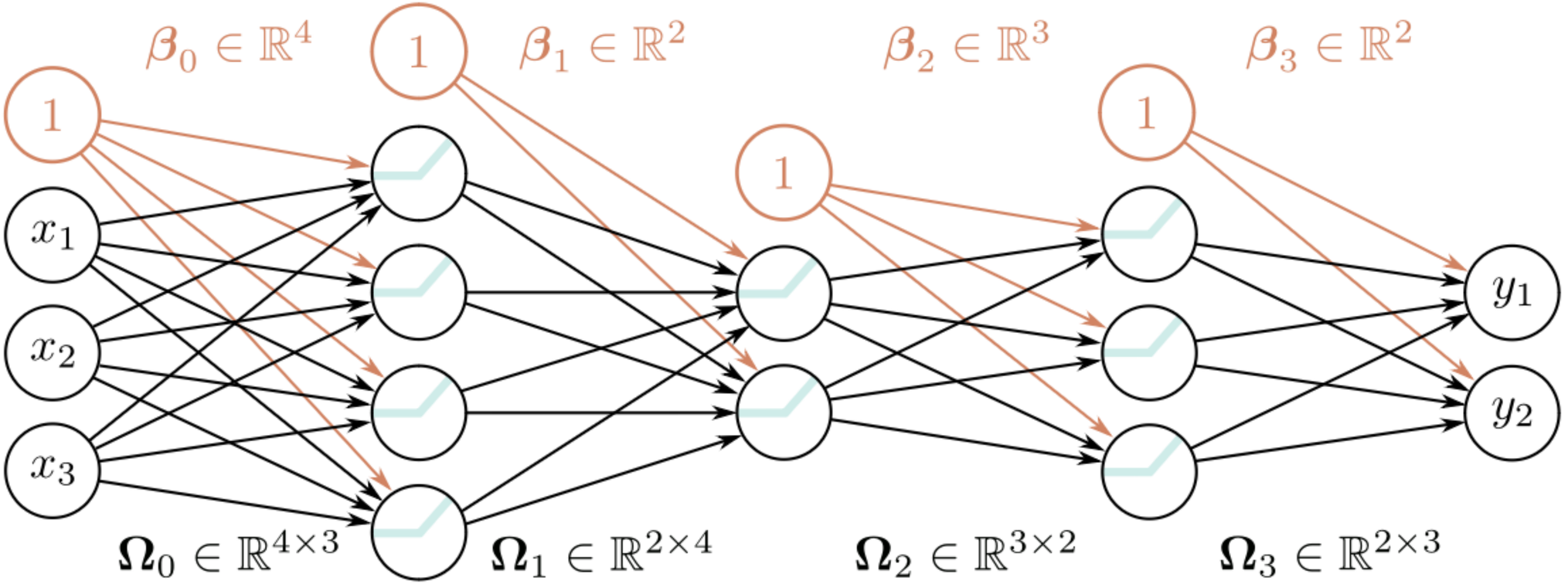
Output, $\mathbf{y}$

$D_o = 2$

Man

By

Upon



Input, $\mathbf{x}$

$D_i = 3$

Hidden layer, $\mathbf{h}_1$

$D_1 = 4$

Hidden layer, $\mathbf{h}_2$

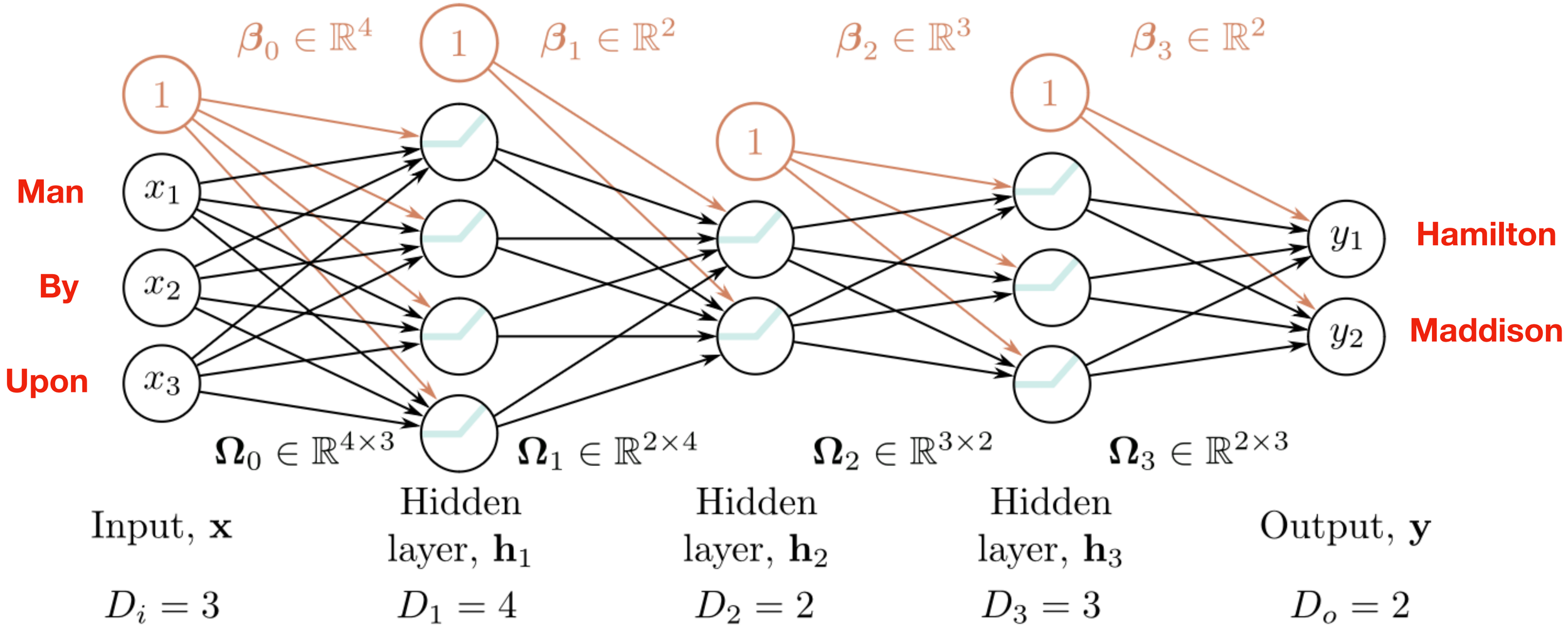
$D_2 = 2$

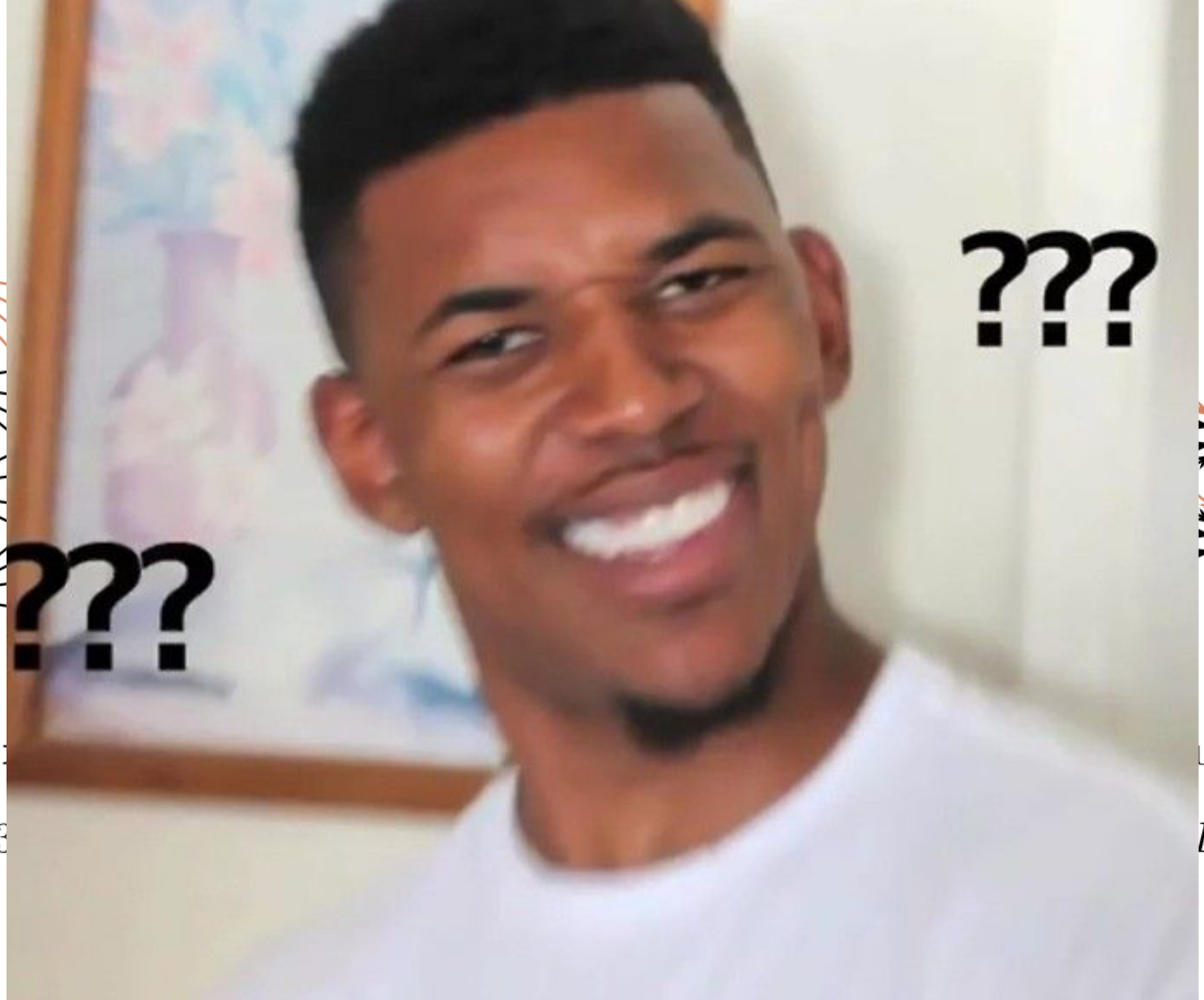
Hidden layer, $\mathbf{h}_3$

$D_3 = 3$

Output, $\mathbf{y}$

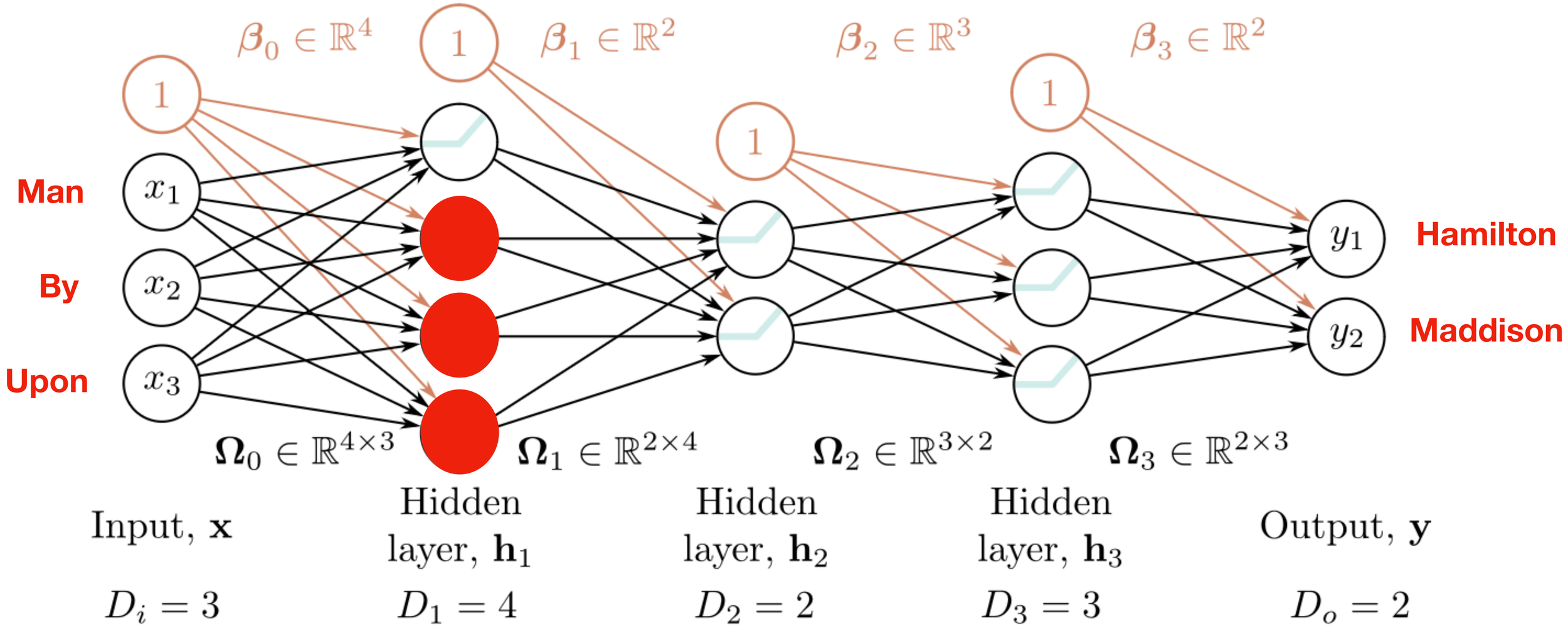
$D_o = 2$

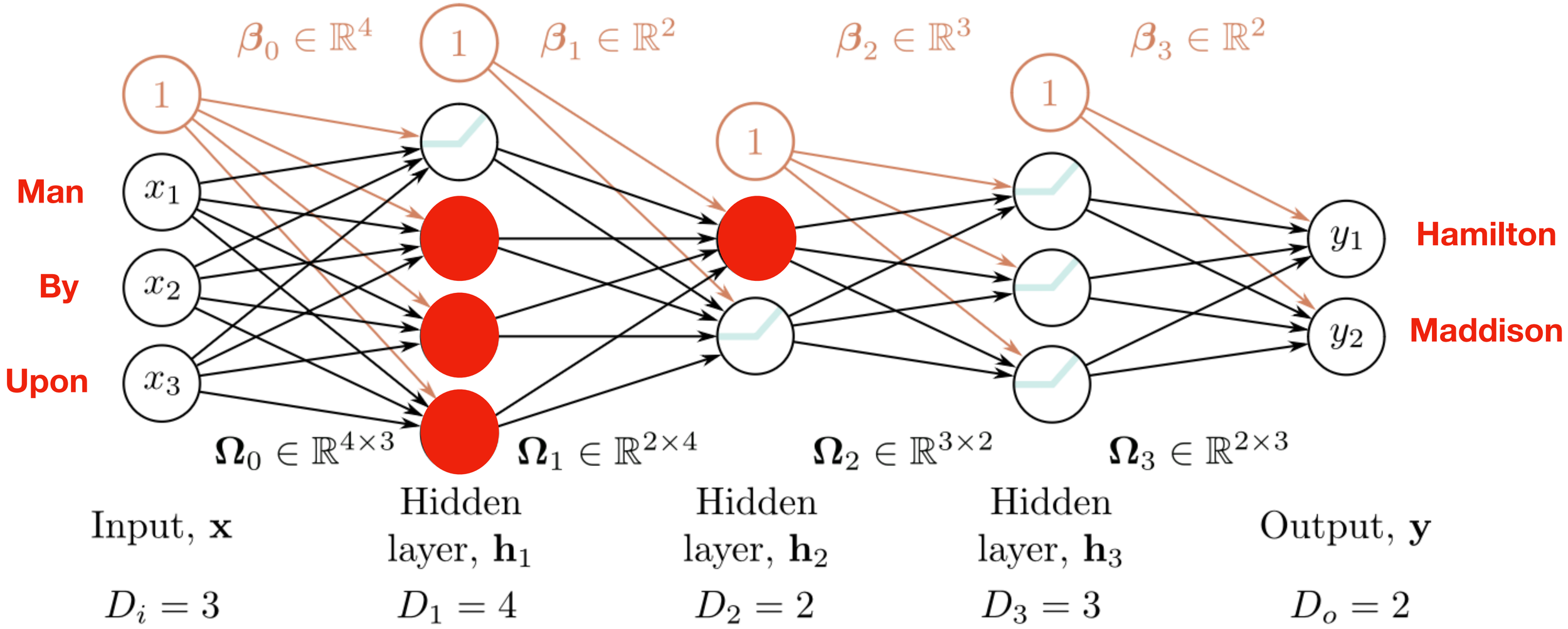

$\boldsymbol{\beta}_0 \in \mathbb{R}^4$
$\boldsymbol{\beta}_1 \in \mathbb{R}^2$
$\boldsymbol{\beta}_2 \in \mathbb{R}^3$
$\boldsymbol{\beta}_3 \in \mathbb{R}^2$
1
x_1
x_2
x_3
y_1
y_2
Man
By
Upon
Hamilton
Maddison
$\boldsymbol{\Omega}_0 \in \mathbb{R}^{4\times3}$
$\boldsymbol{\Omega}_1 \in \mathbb{R}^{2\times4}$
$\boldsymbol{\Omega}_2 \in \mathbb{R}^{3\times2}$
$\boldsymbol{\Omega}_3 \in \mathbb{R}^{2\times3}$
Input, $\mathbf{x}$
$D_i = 3$
Hidden layer, $\mathbf{h}_1$
$D_1 = 4$
Hidden layer, $\mathbf{h}_2$
$D_2 = 2$
Hidden layer, $\mathbf{h}_3$
$D_3 = 3$
Output, $\mathbf{y}$
$D_o = 2$

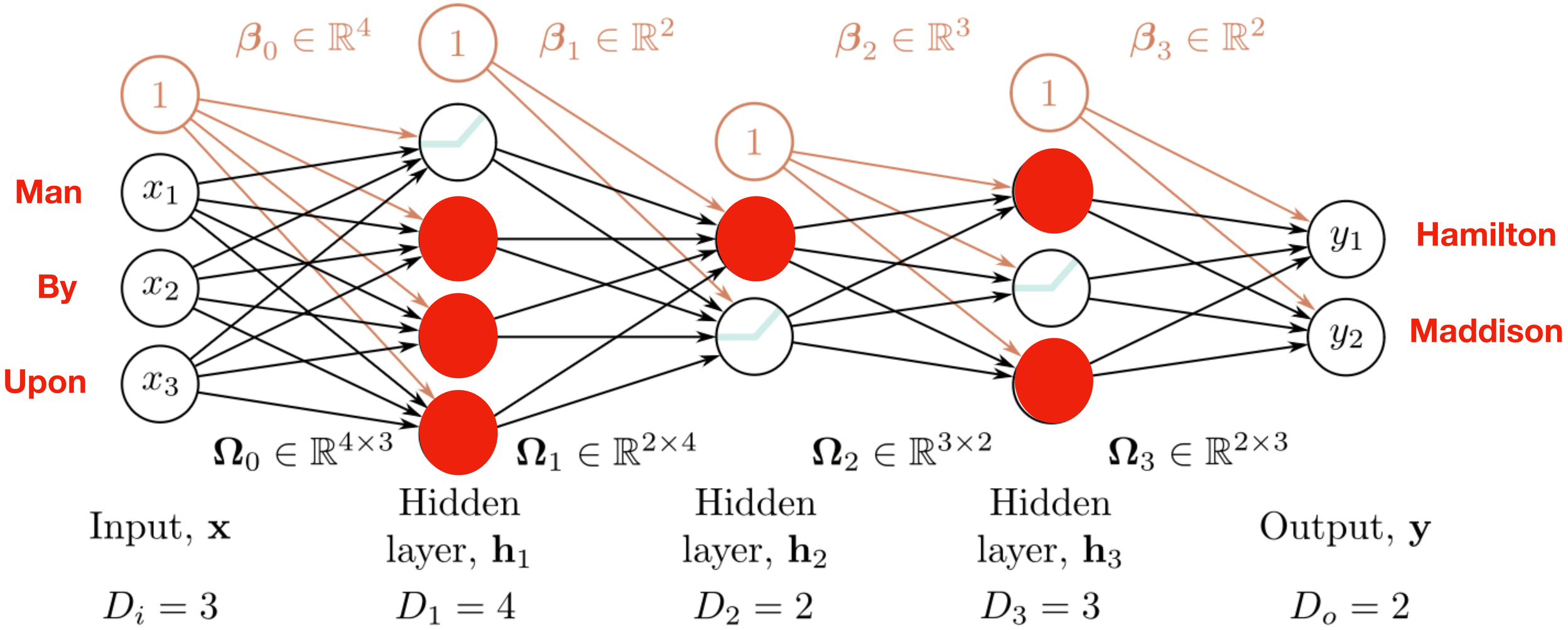


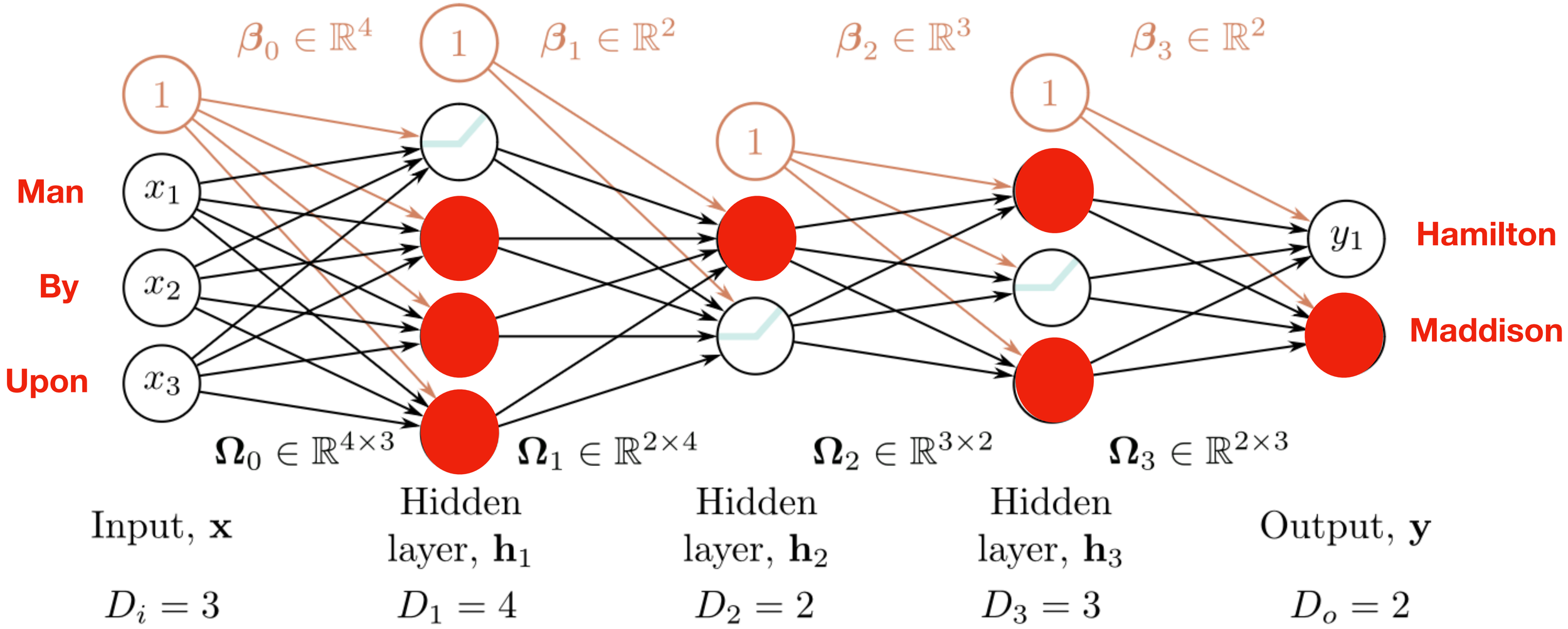
Input, $\mathbf{x}$ utput, $\mathbf{y}$

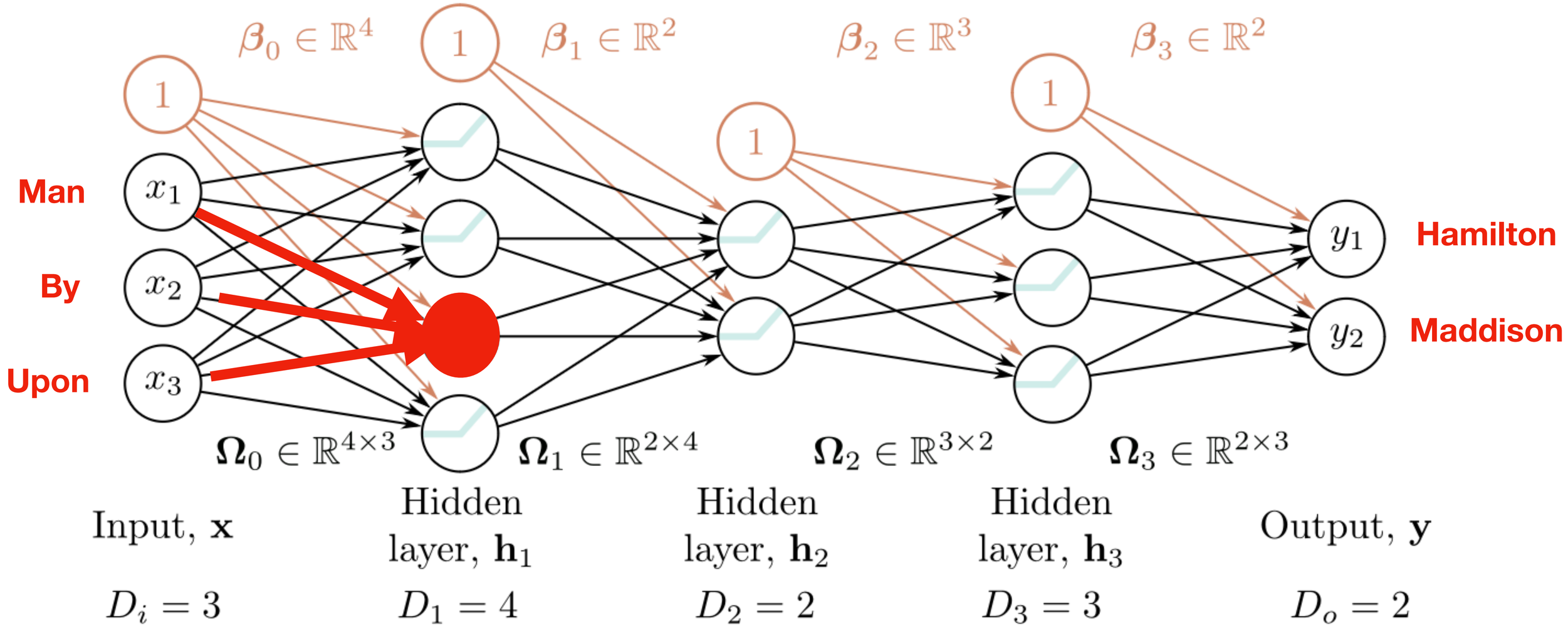
$D_i = 3$ $D_o = 2$

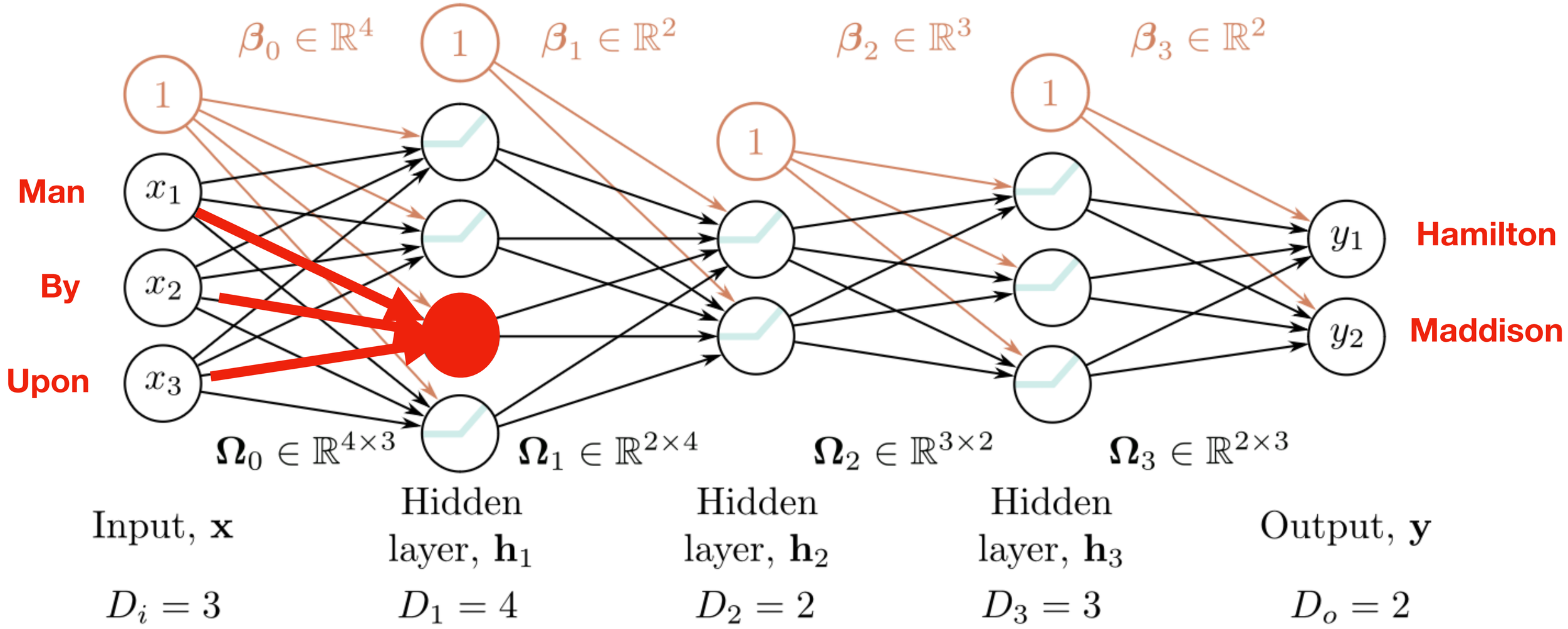

$\beta_0 \in \mathbb{R}^4$
$\beta_1 \in \mathbb{R}^2$
$\beta_2 \in \mathbb{R}^3$
$\beta_3 \in \mathbb{R}^2$
1
1
1
1
x_1
x_2
x_3
y_1
y_2
Man
By
Upon
Hamilton
Maddison
$\Omega_0 \in \mathbb{R}^{4\times3}$
$\Omega_1 \in \mathbb{R}^{2\times4}$
$\Omega_2 \in \mathbb{R}^{3\times2}$
$\Omega_3 \in \mathbb{R}^{2\times3}$
Input, $\mathbf{x}$
$D_i = 3$
Hidden layer, $\mathbf{h}_1$
$D_1 = 4$
Hidden layer, $\mathbf{h}_2$
$D_2 = 2$
Hidden layer, $\mathbf{h}_3$
$D_3 = 3$
Output, $\mathbf{y}$
$D_o = 2$


$\boldsymbol{\beta}_0 \in \mathbb{R}^4$
$\boldsymbol{\beta}_1 \in \mathbb{R}^2$
$\boldsymbol{\beta}_2 \in \mathbb{R}^3$
$\boldsymbol{\beta}_3 \in \mathbb{R}^2$
1
x_1
x_2
x_3
y_1
y_2
Man
By
Upon
Hamilton
Maddison
$\boldsymbol{\Omega}_0 \in \mathbb{R}^{4\times3}$
$\boldsymbol{\Omega}_1 \in \mathbb{R}^{2\times4}$
$\boldsymbol{\Omega}_2 \in \mathbb{R}^{3\times2}$
$\boldsymbol{\Omega}_3 \in \mathbb{R}^{2\times3}$
Input, $\mathbf{x}$
$D_i = 3$
Hidden layer, $\mathbf{h}_1$
$D_1 = 4$
Hidden layer, $\mathbf{h}_2$
$D_2 = 2$
Hidden layer, $\mathbf{h}_3$
$D_3 = 3$
Output, $\mathbf{y}$
$D_o = 2$


$\beta_0 \in \mathbb{R}^4$
$\beta_1 \in \mathbb{R}^2$
$\beta_2 \in \mathbb{R}^3$
$\beta_3 \in \mathbb{R}^2$
1
1
1
1
x_1
x_2
x_3
y_1
y_2
Man
By
Upon
Hamilton
Maddison
$\Omega_0 \in \mathbb{R}^{4\times3}$
$\Omega_1 \in \mathbb{R}^{2\times4}$
$\Omega_2 \in \mathbb{R}^{3\times2}$
$\Omega_3 \in \mathbb{R}^{2\times3}$
Input, $\mathbf{x}$
$D_i = 3$
Hidden layer, $\mathbf{h}_1$
$D_1 = 4$
Hidden layer, $\mathbf{h}_2$
$D_2 = 2$
Hidden layer, $\mathbf{h}_3$
$D_3 = 3$
Output, $\mathbf{y}$
$D_o = 2$


$\boldsymbol{\beta}_0 \in \mathbb{R}^4$
$\boldsymbol{\beta}_1 \in \mathbb{R}^2$
$\boldsymbol{\beta}_2 \in \mathbb{R}^3$
$\boldsymbol{\beta}_3 \in \mathbb{R}^2$
1
x_1
x_2
x_3
y_1
Man
By
Upon
Hamilton
Maddison
$\boldsymbol{\Omega}_0 \in \mathbb{R}^{4\times3}$
$\boldsymbol{\Omega}_1 \in \mathbb{R}^{2\times4}$
$\boldsymbol{\Omega}_2 \in \mathbb{R}^{3\times2}$
$\boldsymbol{\Omega}_3 \in \mathbb{R}^{2\times3}$
Input, $\mathbf{x}$
$D_i = 3$
Hidden layer, $\mathbf{h}_1$
$D_1 = 4$
Hidden layer, $\mathbf{h}_2$
$D_2 = 2$
Hidden layer, $\mathbf{h}_3$
$D_3 = 3$
Output, $\mathbf{y}$
$D_o = 2$


$\beta_0 \in \mathbb{R}^4$
$\beta_1 \in \mathbb{R}^2$
$\beta_2 \in \mathbb{R}^3$
$\beta_3 \in \mathbb{R}^2$
1
x_1
x_2
x_3
y_1
y_2
Man
By
Upon
Hamilton
Maddison
$\Omega_0 \in \mathbb{R}^{4\times3}$
$\Omega_1 \in \mathbb{R}^{2\times4}$
$\Omega_2 \in \mathbb{R}^{3\times2}$
$\Omega_3 \in \mathbb{R}^{2\times3}$
Input, $\mathbf{x}$
$D_i = 3$
Hidden layer, $\mathbf{h}_1$
$D_1 = 4$
Hidden layer, $\mathbf{h}_2$
$D_2 = 2$
Hidden layer, $\mathbf{h}_3$
$D_3 = 3$
Output, $\mathbf{y}$
$D_o = 2$


$\beta_0 \in \mathbb{R}^4$
$\beta_1 \in \mathbb{R}^2$
$\beta_2 \in \mathbb{R}^3$
$\beta_3 \in \mathbb{R}^2$
1
x_1
x_2
x_3
y_1
y_2
Man
By
Upon
Hamilton
Maddison
$\Omega_0 \in \mathbb{R}^{4\times3}$
$\Omega_1 \in \mathbb{R}^{2\times4}$
$\Omega_2 \in \mathbb{R}^{3\times2}$
$\Omega_3 \in \mathbb{R}^{2\times3}$
Input, $\mathbf{x}$
$D_i = 3$
Hidden layer, $\mathbf{h}_1$
$D_1 = 4$
Hidden layer, $\mathbf{h}_2$
$D_2 = 2$
Hidden layer, $\mathbf{h}_3$
$D_3 = 3$
Output, $\mathbf{y}$
$D_o = 2$

$$w_1 a_1 + w_2 a_2 + w_3 a_3 + \ldots + w_n a_n$$

$$\boxed{w_1 a_1 + w_2 a_2 + w_3 a_3} + \ldots + w_n a_n$$

Activation of a Neuron

$$w_1 a_1 + w_2 a_2 + w_3 a_3 + \ldots + w_n a_n$$

Activation of a Neuron

$$w_1 a_1 + w_2 a_2 + w_3 a_3 + \ldots + w_n a_n \longrightarrow [-\infty; \infty]$$

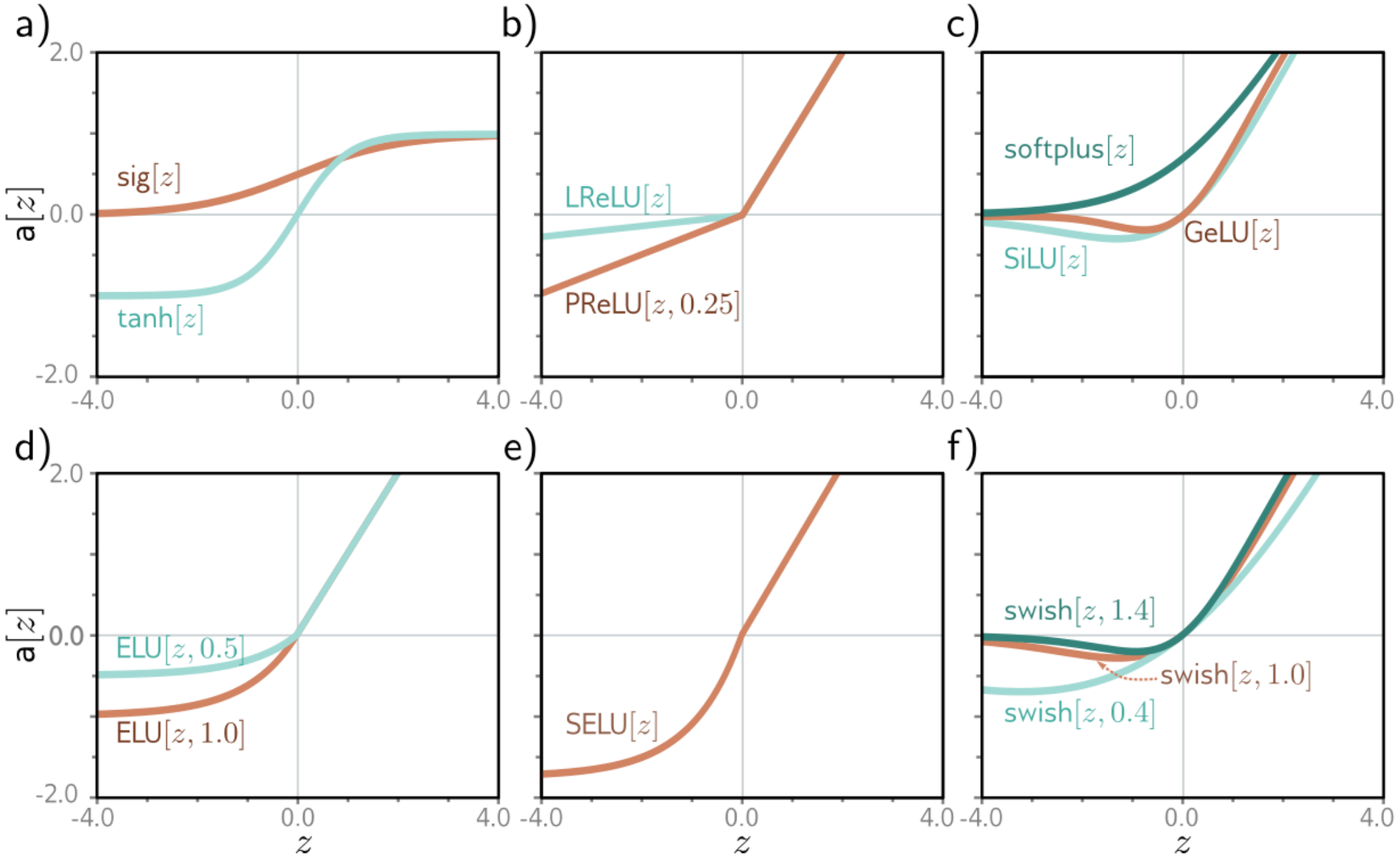
Activation of a Neuron

$$w_1 a_1 + w_2 a_2 + w_3 a_3 + \ldots + w_n a_n \longrightarrow f(\,\cdot\,) \qquad [-\infty; \infty]$$

Activation of a Neuron

$$w_1 a_1 + w_2 a_2 + w_3 a_3 + \ldots + w_n a_n \longrightarrow f(\,\cdot\,) \longrightarrow$$ Something more Meaningful

Activation of a Neuron



Prince, 2023

Figure 3.13 Activation functions. a) Logistic sigmoid and tanh functions. b) Leaky ReLU and parametric ReLU with parameter 0.25. c) SoftPlus, Gaussian error linear unit, and sigmoid linear unit. d) Exponential linear unit with parameters 0.5 and 1.0, e) Scaled exponential linear unit. f) Swish with parameters 0.4, 1.0, and 1.4.

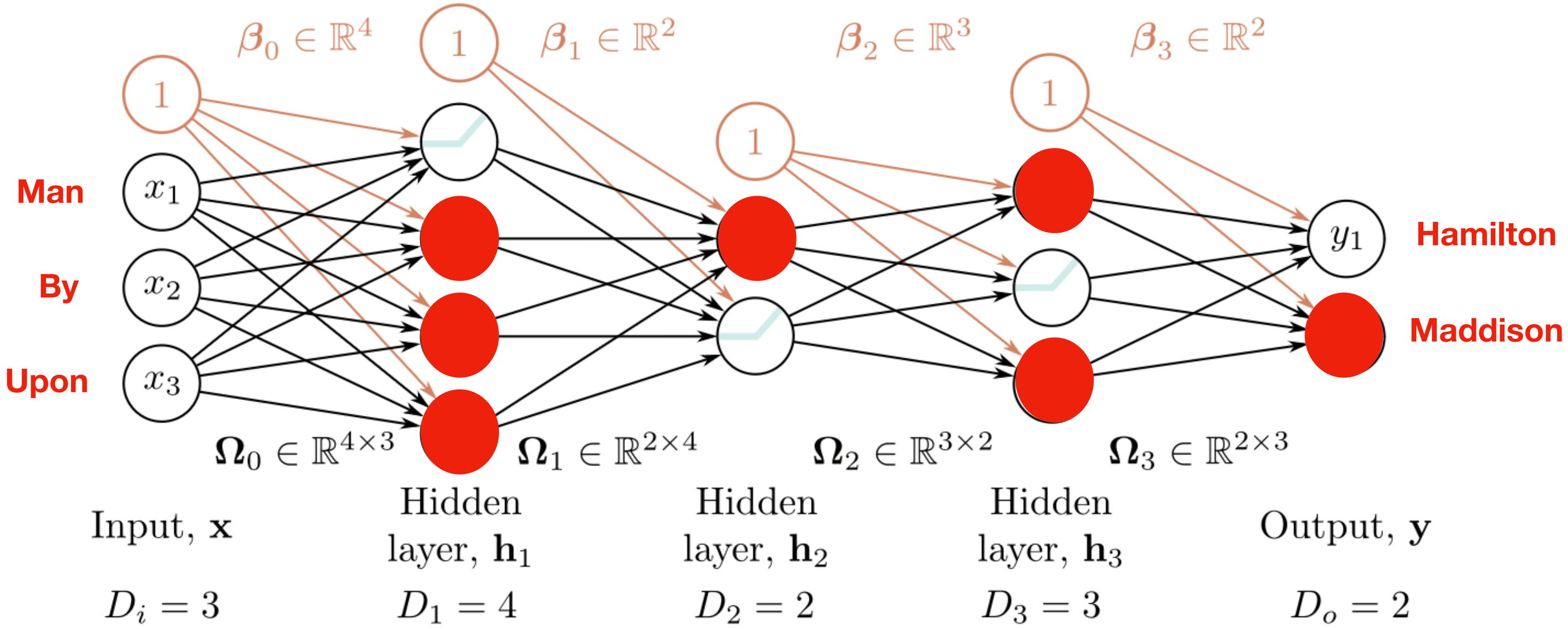
Activation of a Neuron

$$\sigma(w_1 a_1 + w_2 a_2 + \ldots + w_n a_n + \beta)$$

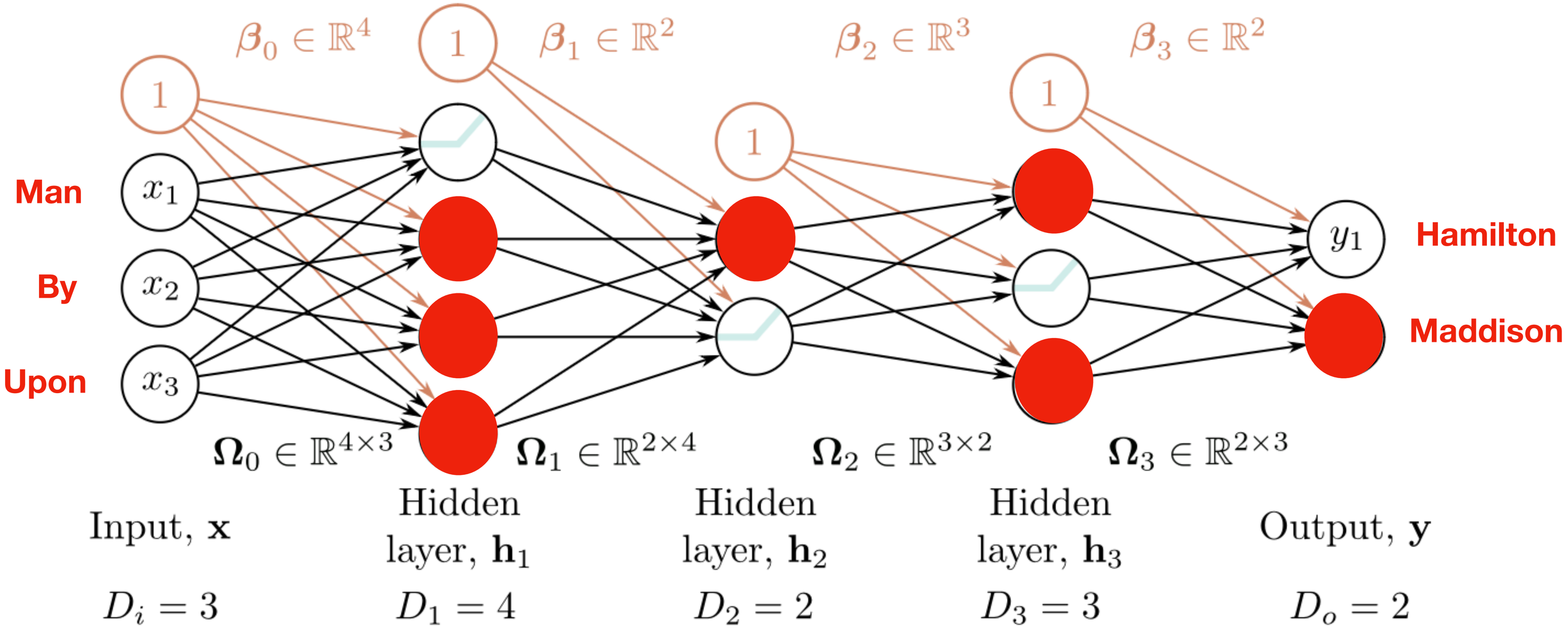
$$f(W^T A + \beta)$$

Activation of a Neuron

$$\sigma(w_1 a_1 + w_2 a_2 + \ldots + w_n a_n + \beta)$$

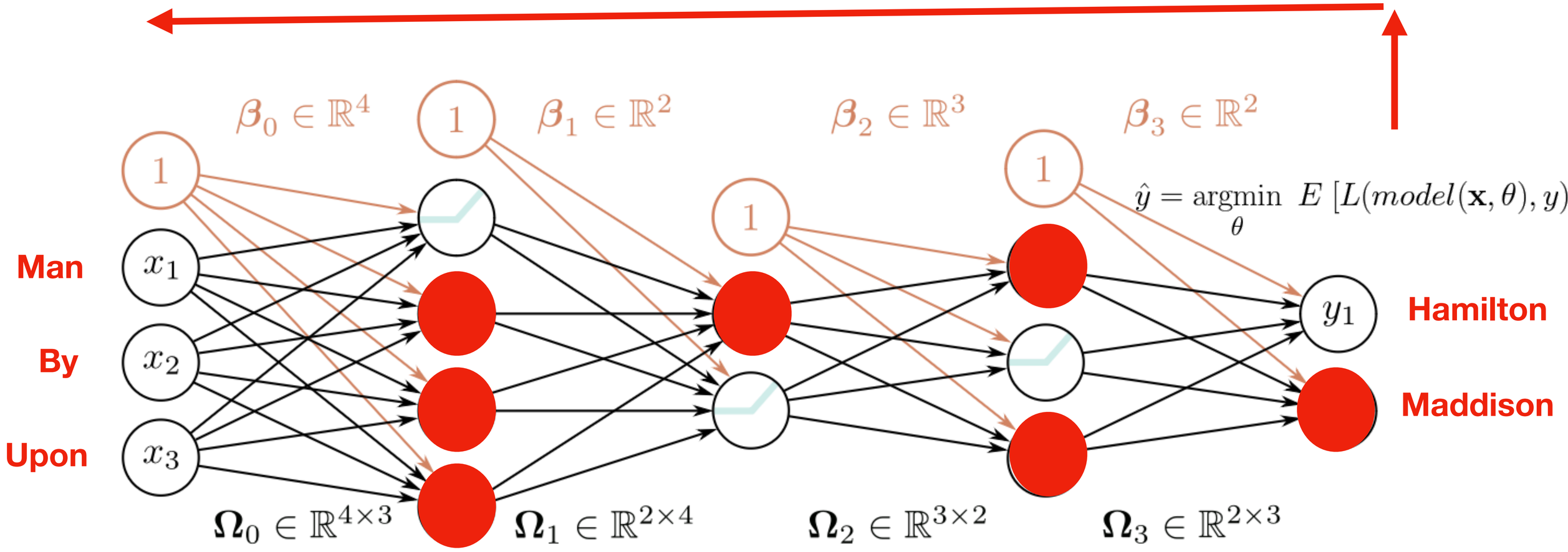

$\boldsymbol{\beta}_0 \in \mathbb{R}^4$
$\boldsymbol{\beta}_1 \in \mathbb{R}^2$
$\boldsymbol{\beta}_2 \in \mathbb{R}^3$
$\boldsymbol{\beta}_3 \in \mathbb{R}^2$
1
x_1
x_2
x_3
y_1
Man
By
Upon
Hamilton
Maddison
$\boldsymbol{\Omega}_0 \in \mathbb{R}^{4\times3}$
$\boldsymbol{\Omega}_1 \in \mathbb{R}^{2\times4}$
$\boldsymbol{\Omega}_2 \in \mathbb{R}^{3\times2}$
$\boldsymbol{\Omega}_3 \in \mathbb{R}^{2\times3}$
Input, $\mathbf{x}$
$D_i = 3$
Hidden layer, $\mathbf{h}_1$
$D_1 = 4$
Hidden layer, $\mathbf{h}_2$
$D_2 = 2$
Hidden layer, $\mathbf{h}_3$
$D_3 = 3$
Output, $\mathbf{y}$
$D_o = 2$

Random Assignment of Weights



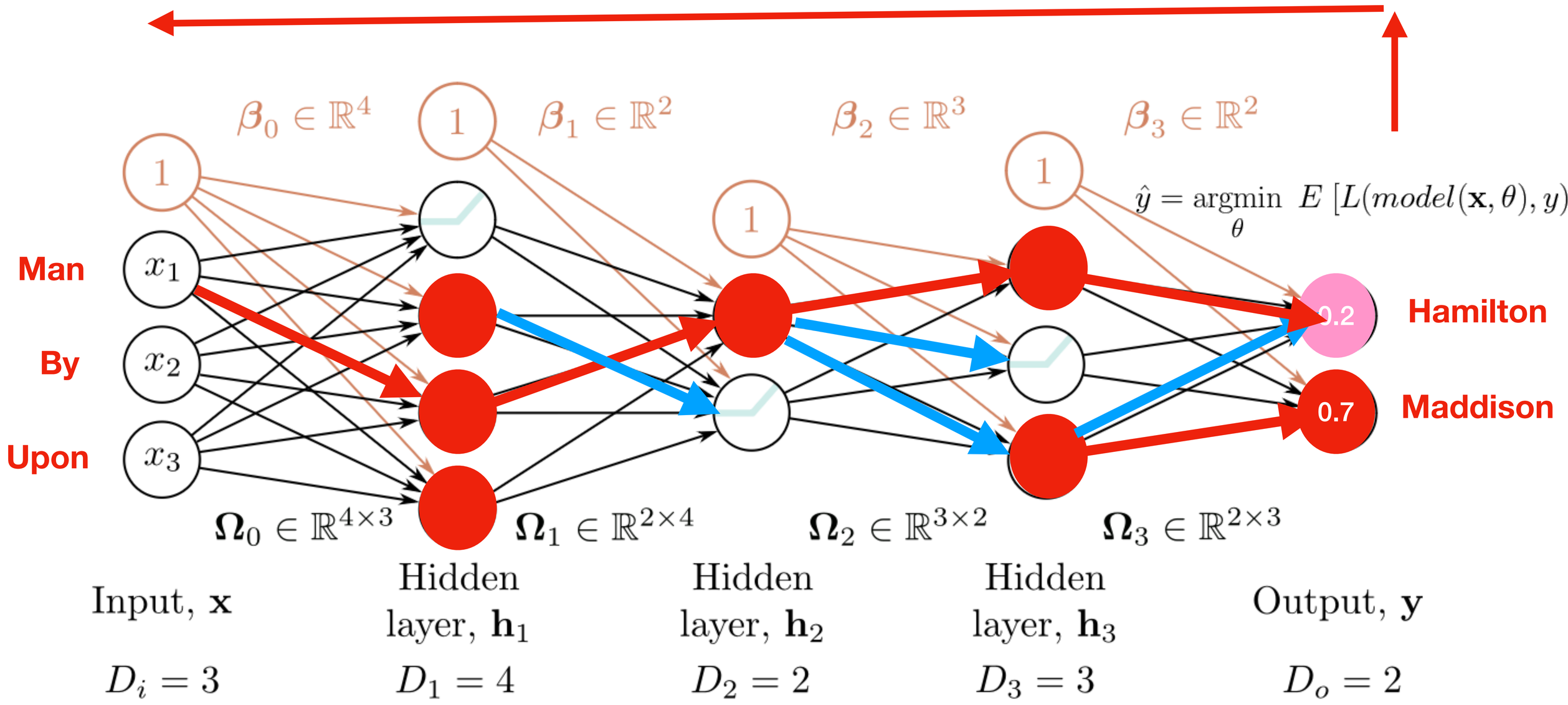
Loss Functions

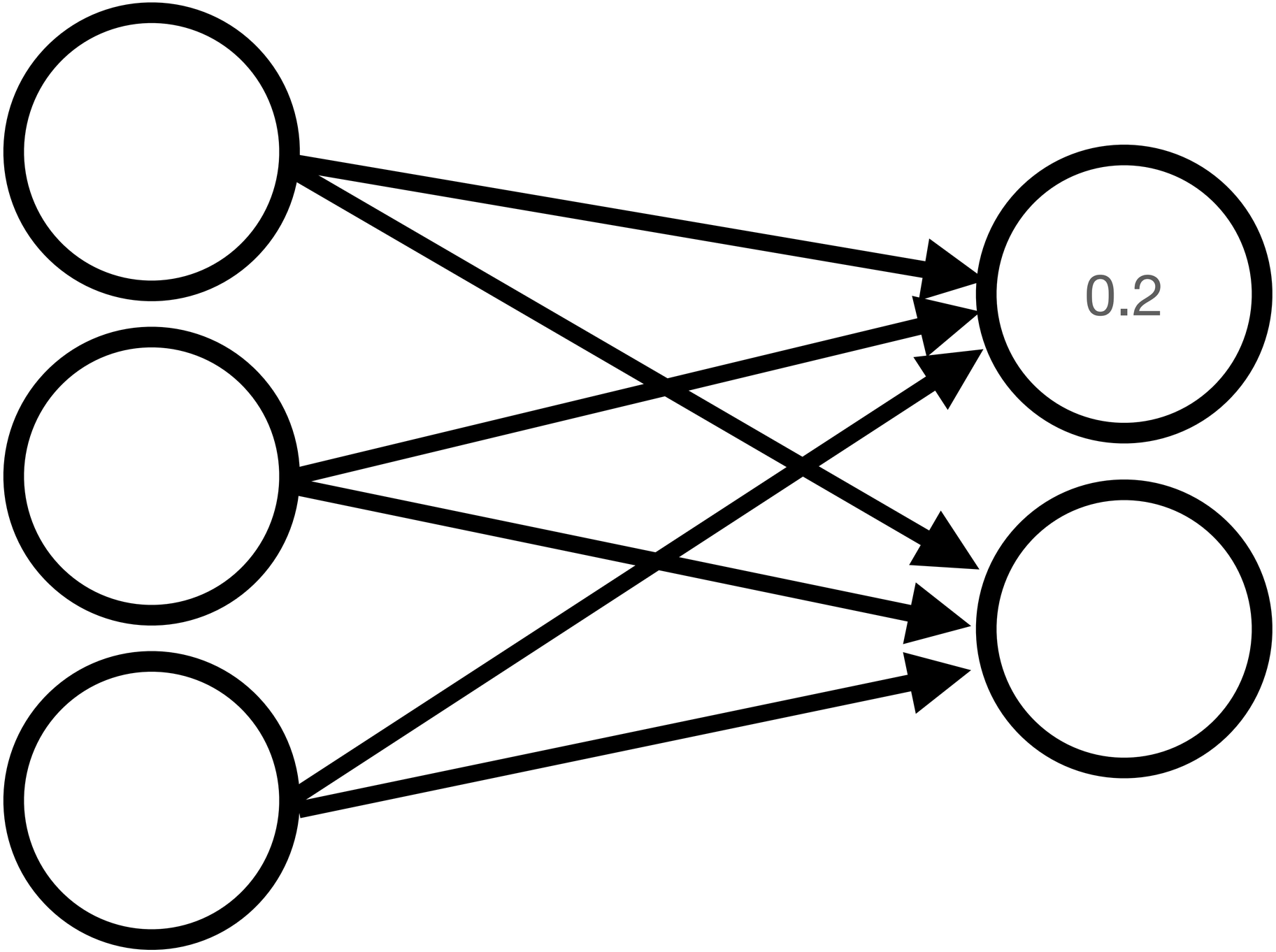
$$\hat{y} = \underset{\theta}{\operatorname{argmin}} \; E\left[L(model(\mathbf{x}, \theta), y)\right]$$

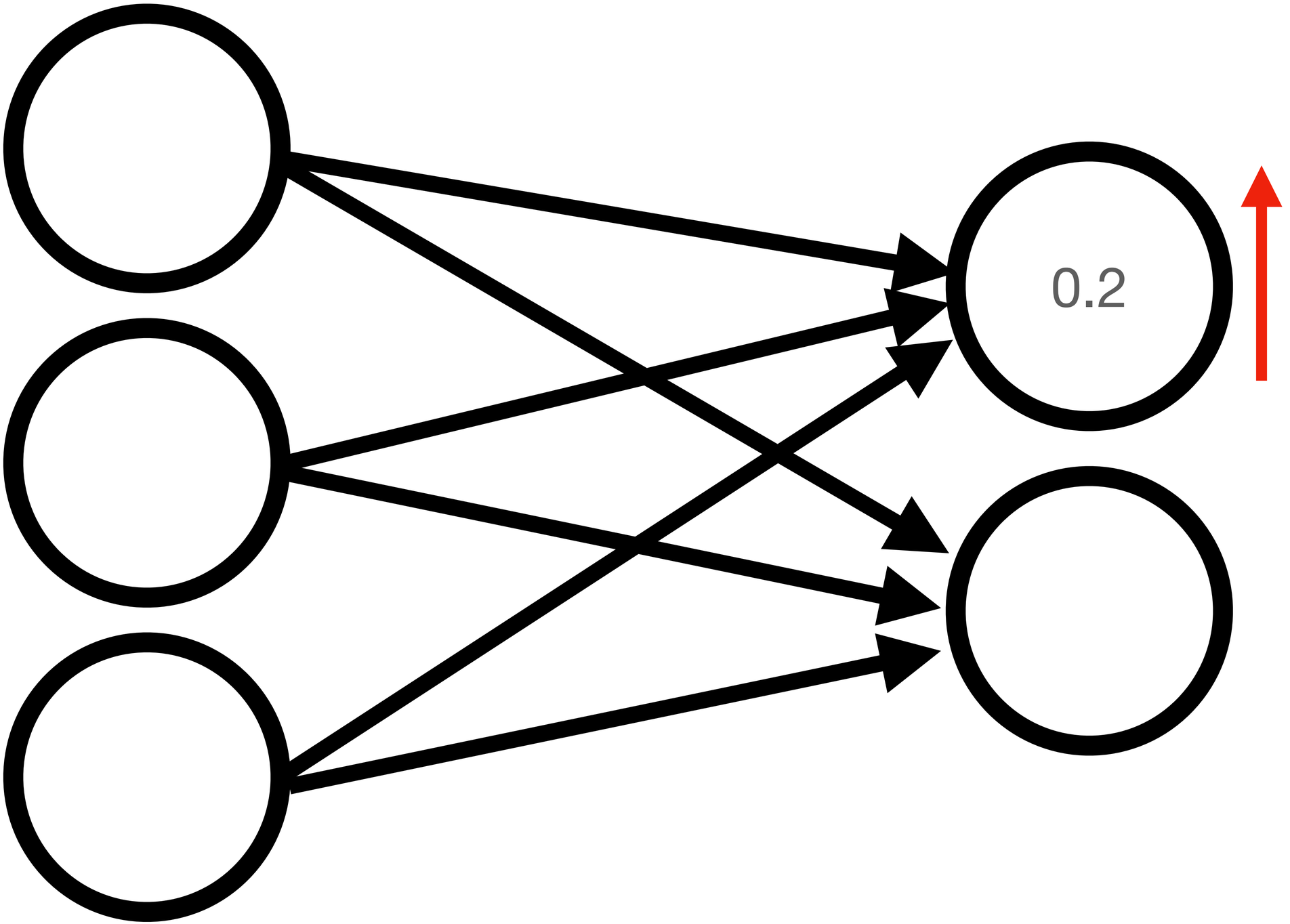

$\beta_0 \in \mathbb{R}^4$
$\beta_1 \in \mathbb{R}^2$
$\beta_2 \in \mathbb{R}^3$
$\beta_3 \in \mathbb{R}^2$
1
x_1
x_2
x_3
$\hat{y} = \underset{\theta}{\text{argmin}}\ E\left[L(model(\mathbf{x}, \theta), y)\right]$
y_1
Man
By
Upon
Hamilton
Maddison
$\mathbf{\Omega}_0 \in \mathbb{R}^{4\times3}$
$\mathbf{\Omega}_1 \in \mathbb{R}^{2\times4}$
$\mathbf{\Omega}_2 \in \mathbb{R}^{3\times2}$
$\mathbf{\Omega}_3 \in \mathbb{R}^{2\times3}$
Input, $\mathbf{x}$
$D_i = 3$
Hidden layer, $\mathbf{h}_1$
$D_1 = 4$
Hidden layer, $\mathbf{h}_2$
$D_2 = 2$
Hidden layer, $\mathbf{h}_3$
$D_3 = 3$
Output, $\mathbf{y}$
$D_o = 2$

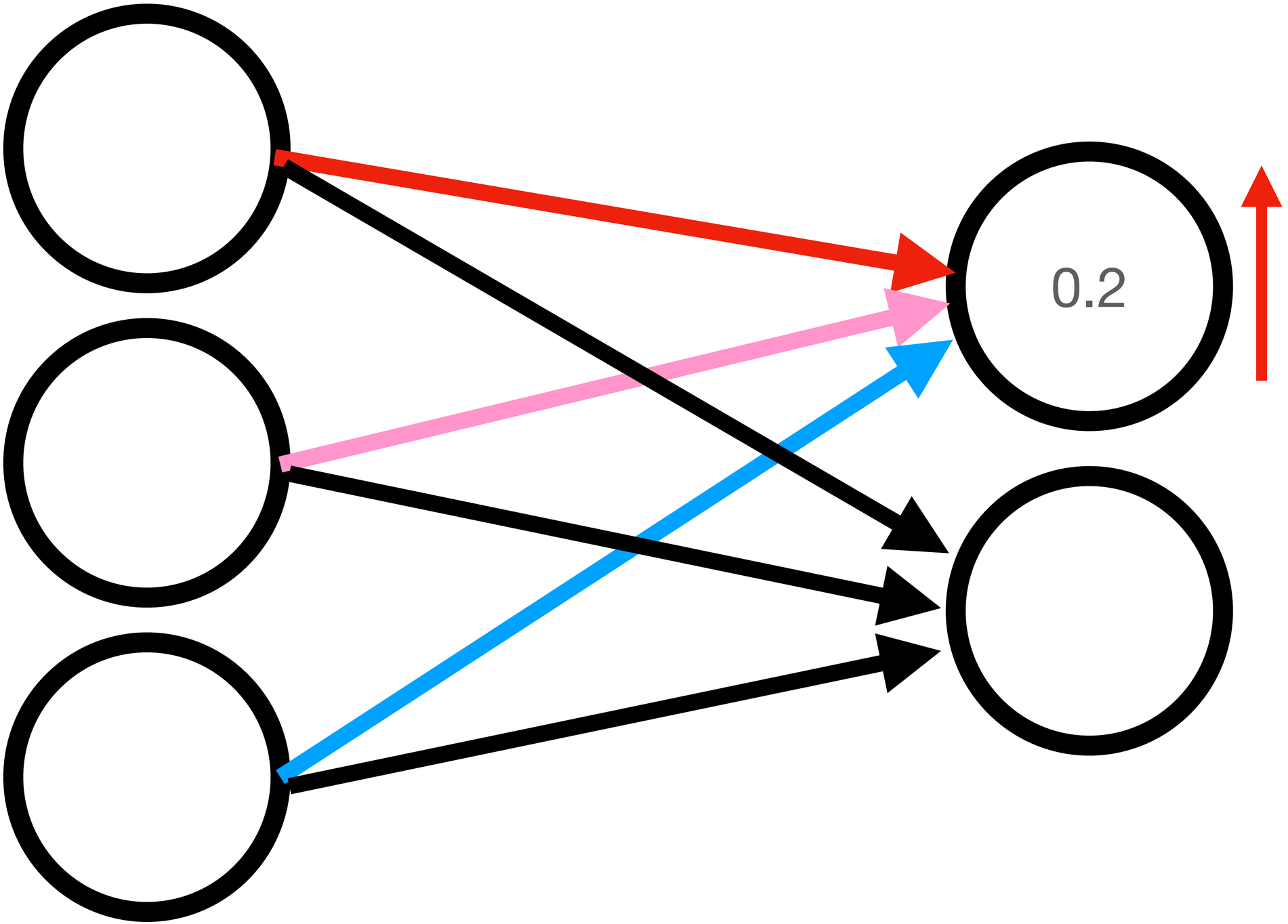
Backpropagation

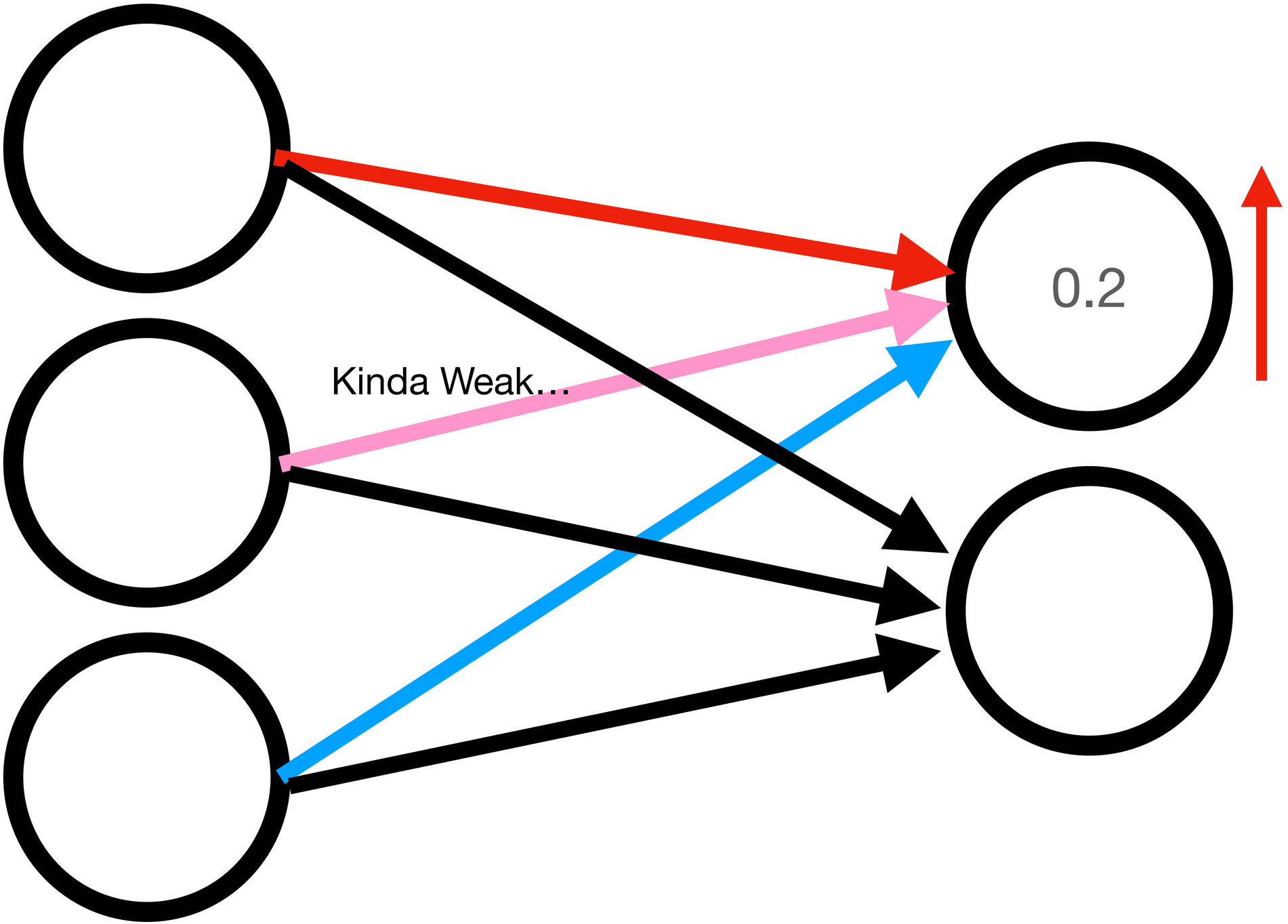
- We need to propagate the gradient back through the network
- Evaluate the gradient of a loss function with the respect to weights
- Recursive application of Chain Rule

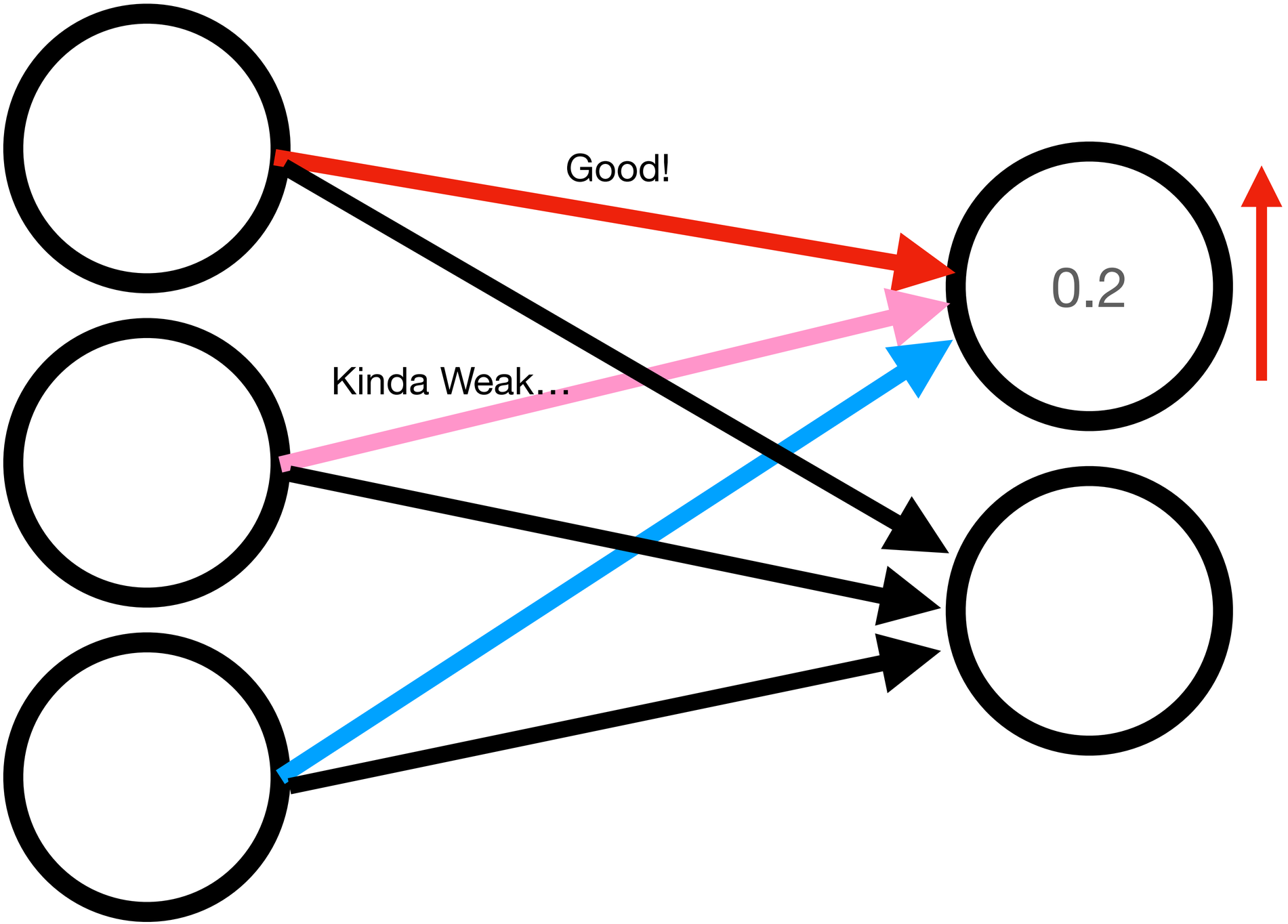

$\beta_0 \in \mathbb{R}^4$
$\beta_1 \in \mathbb{R}^2$
$\beta_2 \in \mathbb{R}^3$
$\beta_3 \in \mathbb{R}^2$
1
1
1
1
$\hat{y} = \underset{\theta}{\text{argmin}}\ E\left[L(model(\mathbf{x}, \theta), y)\right]$
Man
By
Upon
x_1
x_2
x_3
0.2
0.7
Hamilton
Maddison
$\mathbf{\Omega}_0 \in \mathbb{R}^{4\times3}$
$\mathbf{\Omega}_1 \in \mathbb{R}^{2\times4}$
$\mathbf{\Omega}_2 \in \mathbb{R}^{3\times2}$
$\mathbf{\Omega}_3 \in \mathbb{R}^{2\times3}$
Input, $\mathbf{x}$
$D_i = 3$
Hidden layer, $\mathbf{h}_1$
$D_1 = 4$
Hidden layer, $\mathbf{h}_2$
$D_2 = 2$
Hidden layer, $\mathbf{h}_3$
$D_3 = 3$
Output, $\mathbf{y}$
$D_o = 2$

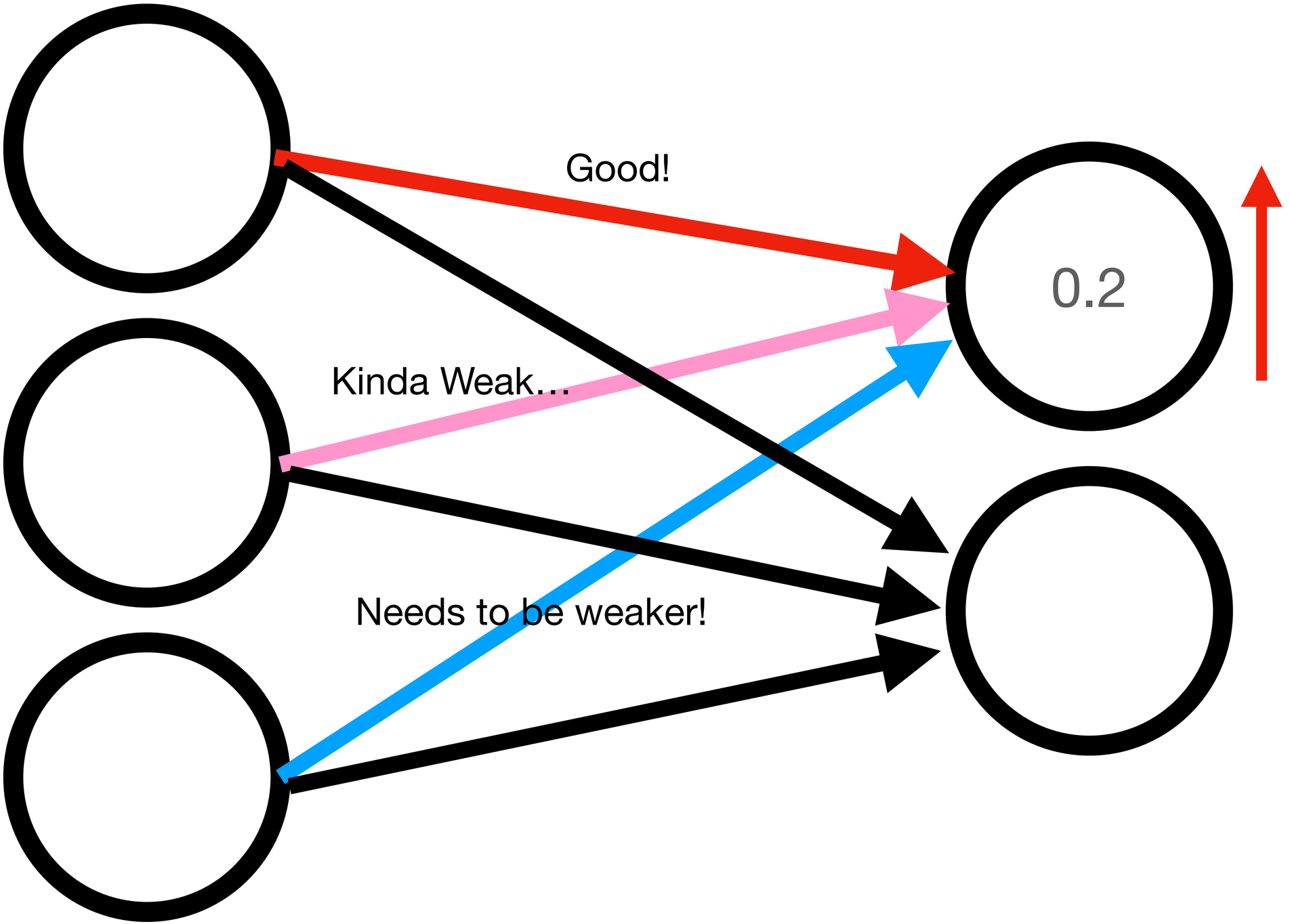

0.2

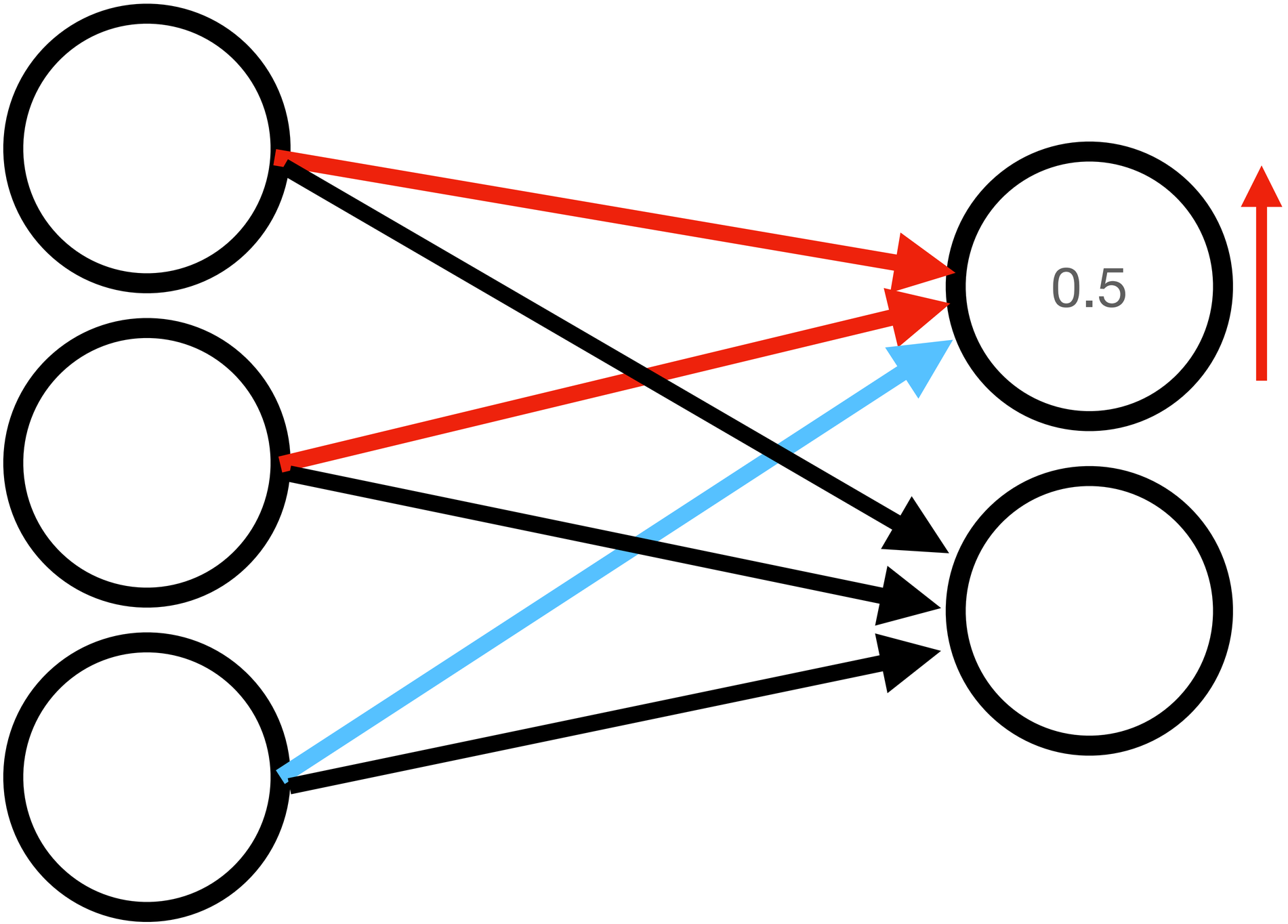

0.2

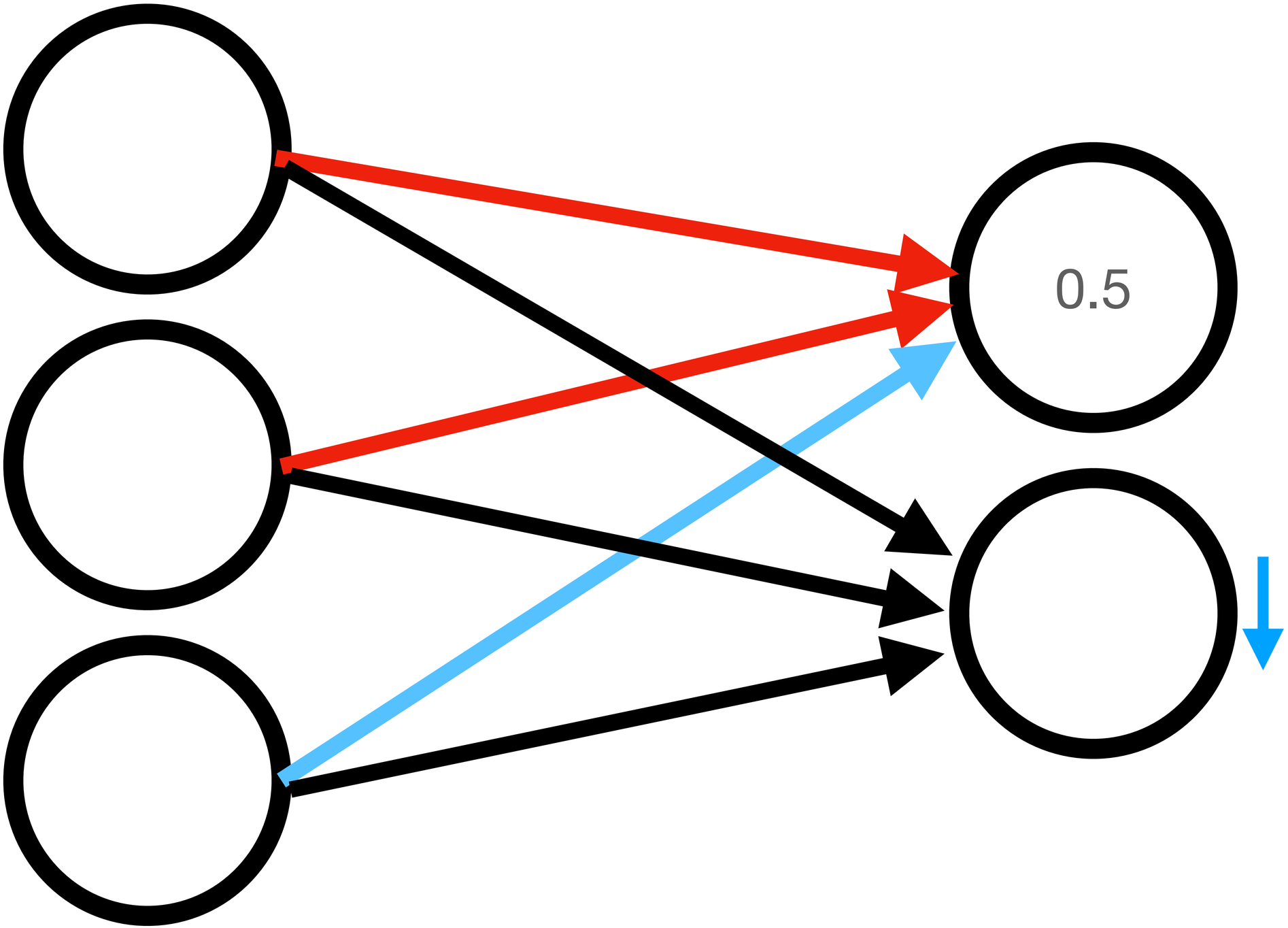

0.2

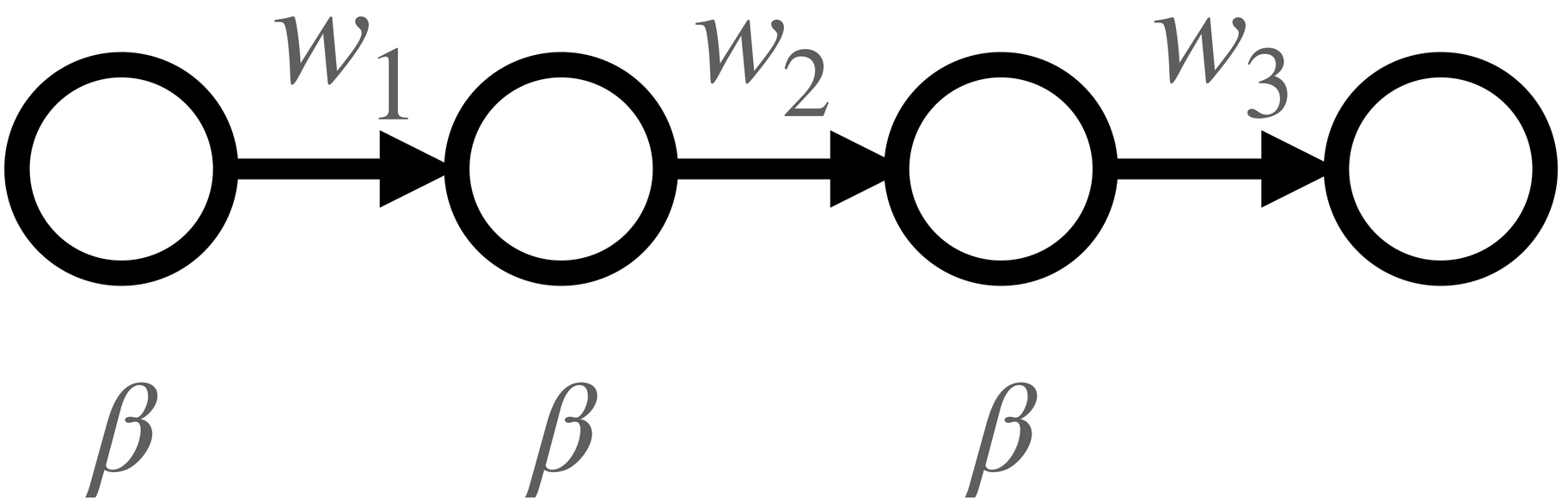

0.2
Kinda Weak…


Good!
Kinda Weak…
0.2

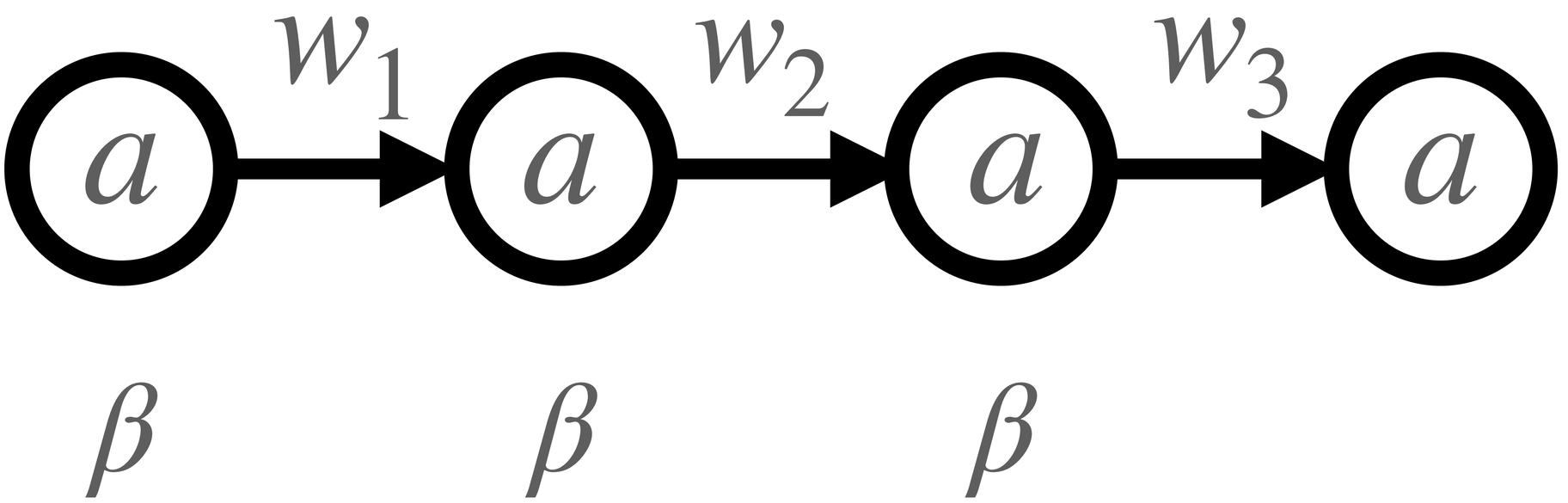

Good!
0.2
Kinda Weak…
Needs to be weaker!


0.5

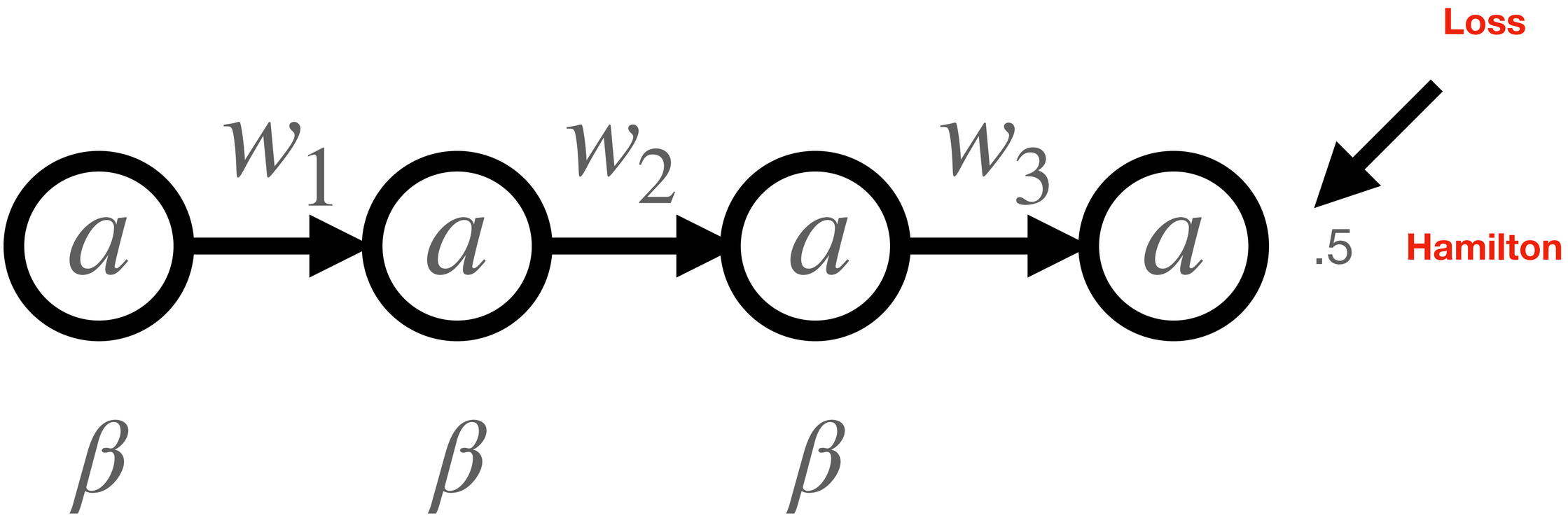

0.5

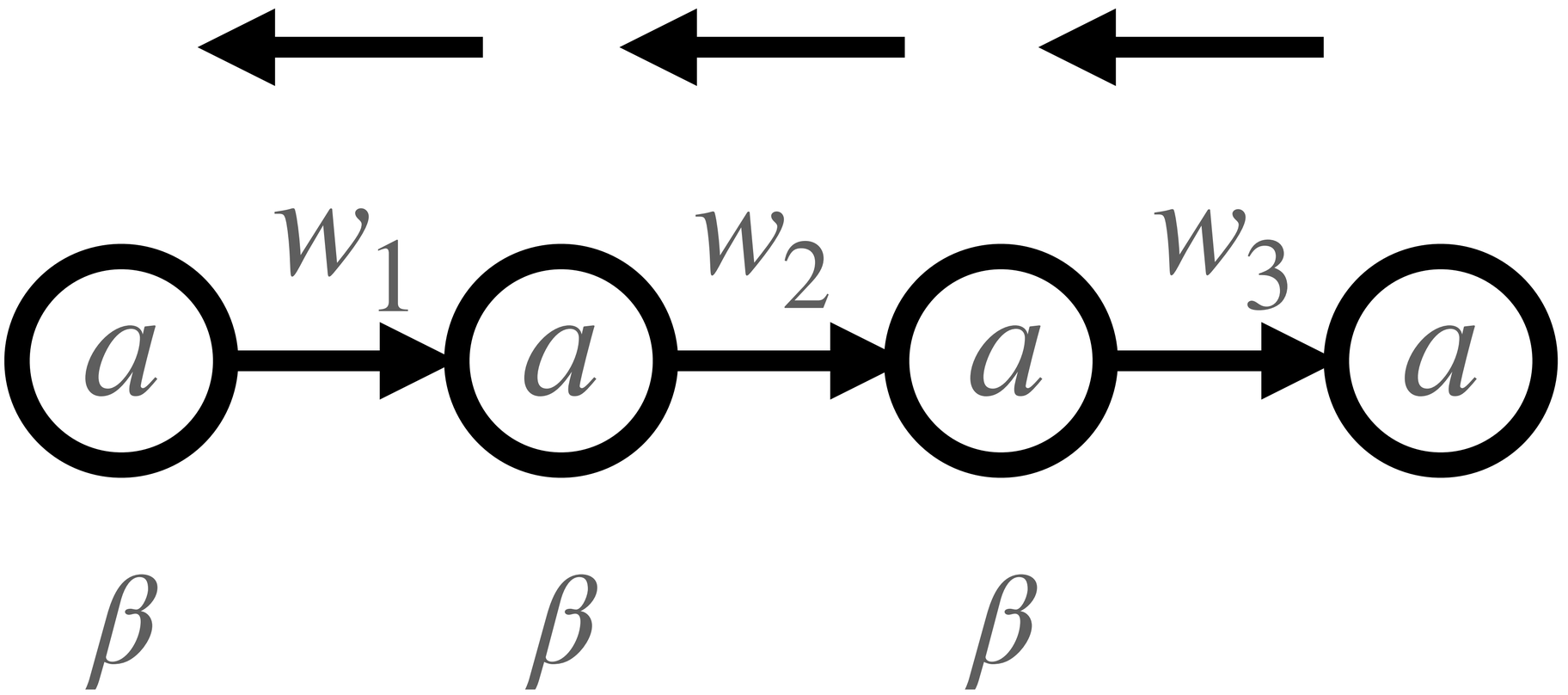


Hamilton



Hamilton


Loss
w_1
w_2
w_3
a
a
a
a
.5
Hamilton
β
β
β



.5 **Hamilton**

Transformer Models

Language Model

- probability distribution over a sequence of words
- e.g., Markov Chains language model

Transformer Models

- A lot more complicated than what we've seen…
- A language model is a probability distribution over a sequence of words

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Attention is all you need

- 79588 citations

Attention is all you need

- 79588 citations
- In ~ 5 years

Attention is all you need

- 79588 citations
- In ~ 5 years
- Developed for machine translation
- Blows all previous NN architectures (like RNNs, LSTMs, etc.) out of the water

Attention is all you need

- 79588 citations
- In ~ 5 years
- Developed for machine translation
- Blows all previous NN architectures (like RNNs, LSTMs, etc.) out of the water
- Bidirectional Encoder Representation from Transformers
- Generative Pre-trained Transformer

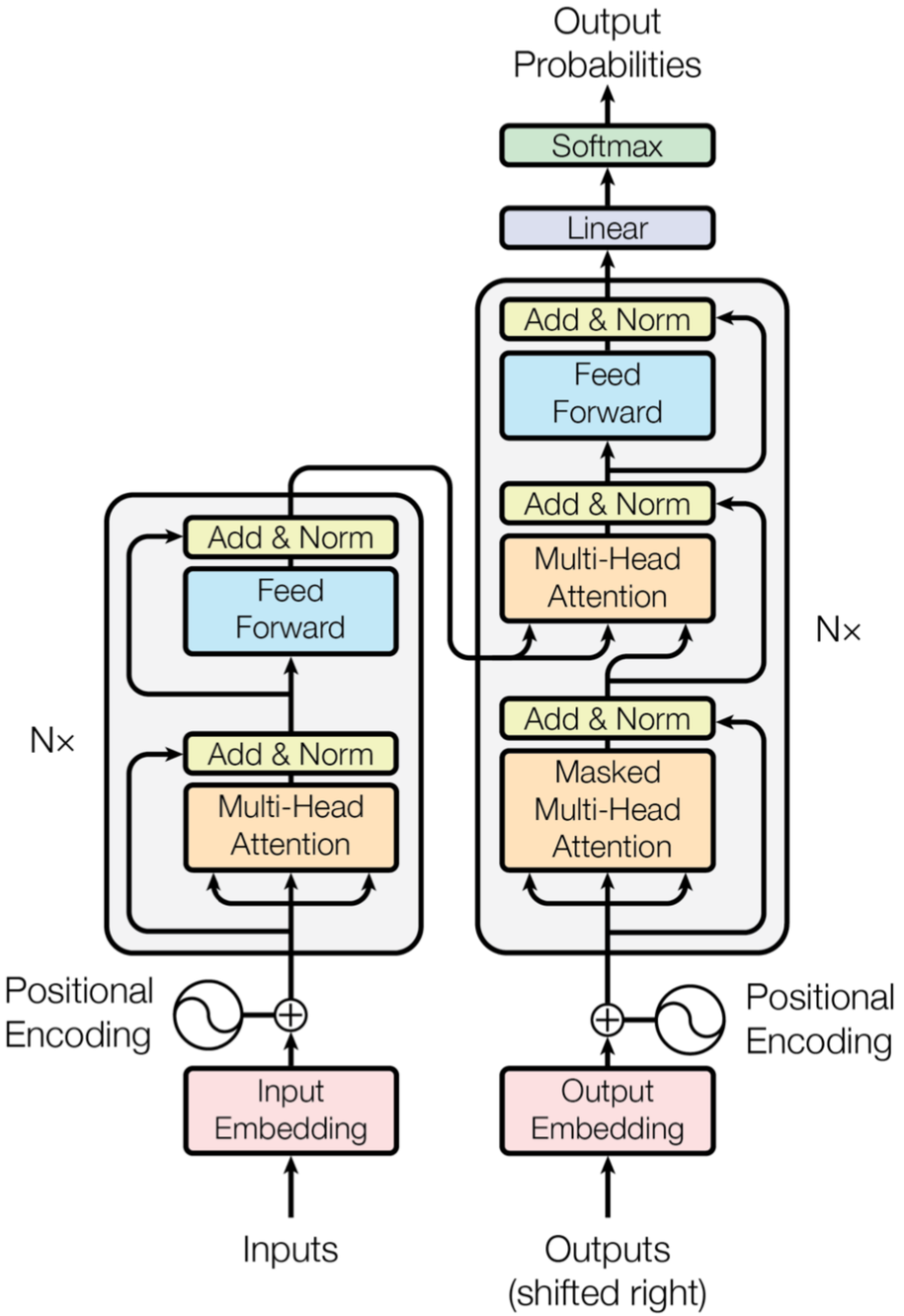
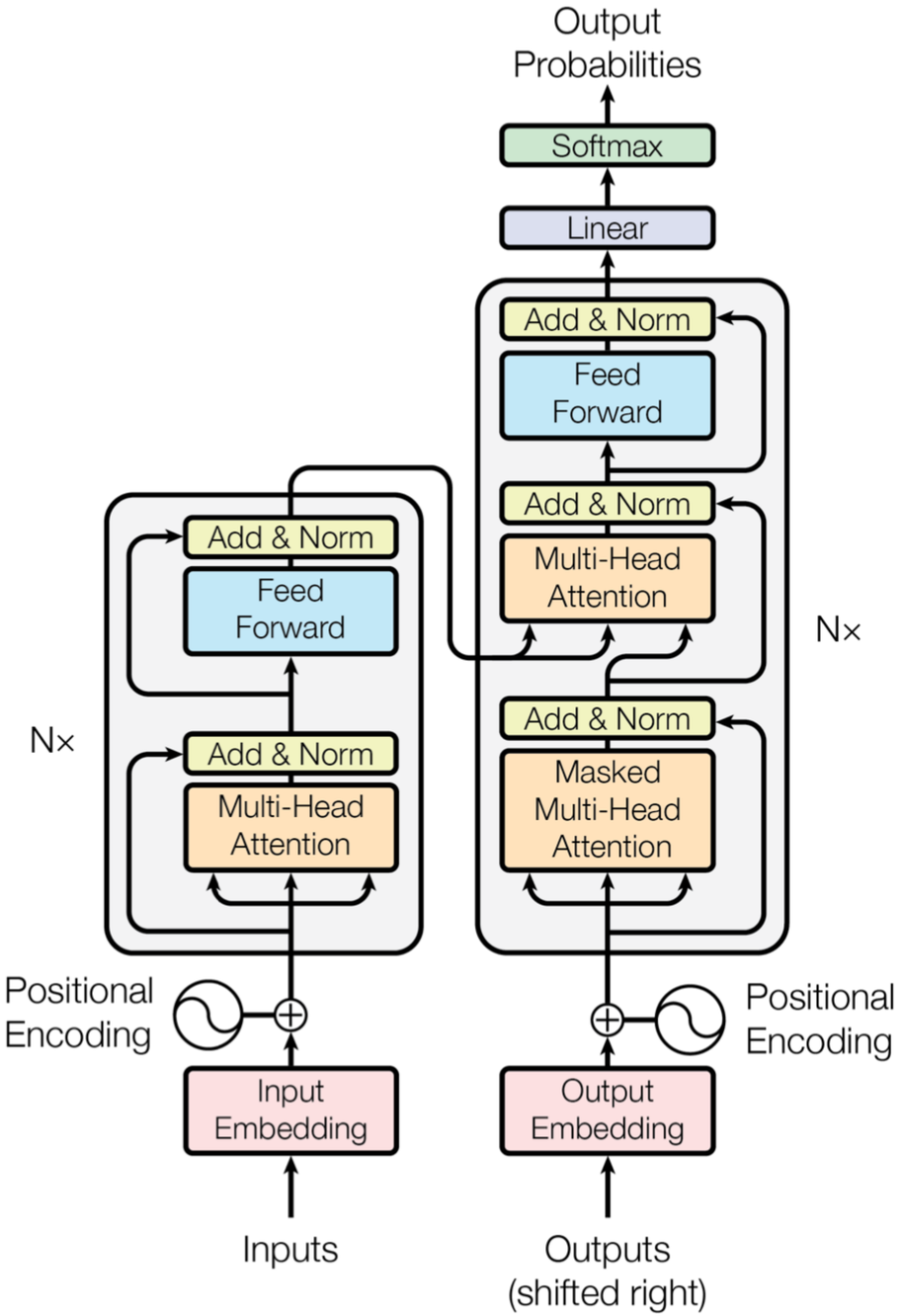


Figure 1: The Transformer - model architecture.

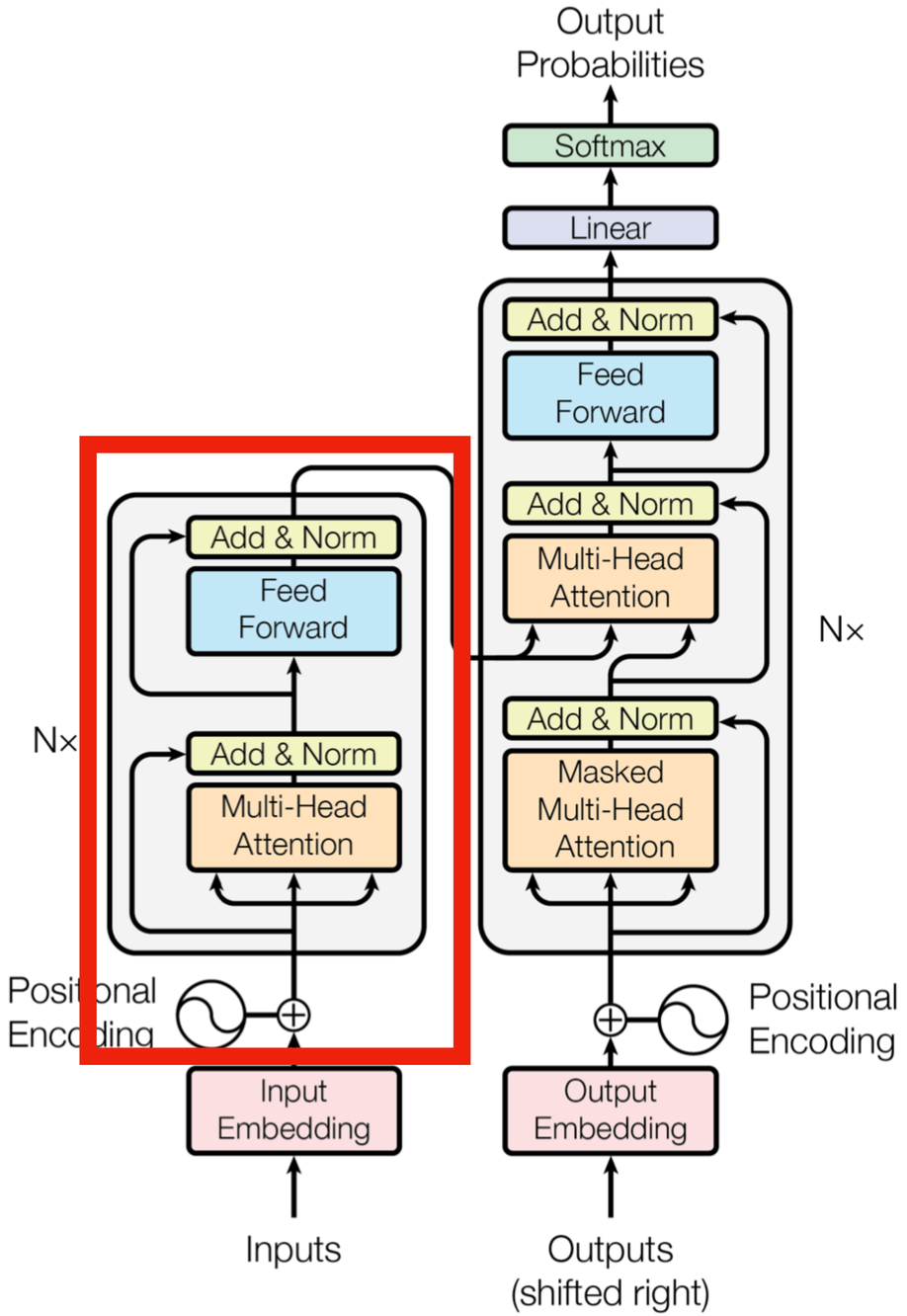
Encoder



Decoder

Figure 1: The Transformer - model architecture.

Encoder



Decoder

Figure 1: The Transformer - model architecture.

Positional Encodings

I love ice-cream

Positional Encodings

I love ice-cream

1 2 3

Self-Attention module

- Handles long-term dependencies
- Pays “attention” to which parts of the input sequence are important
- The self-attention mechanism allows each position in the input sequence to attend to all other positions, weighing the importance of different positions during the computation
- capture long-range dependencies and relationships between words or tokens in the sequence effectively

Encoding for self-attention

- Query
- Key
- Value

Encoding for self-attention

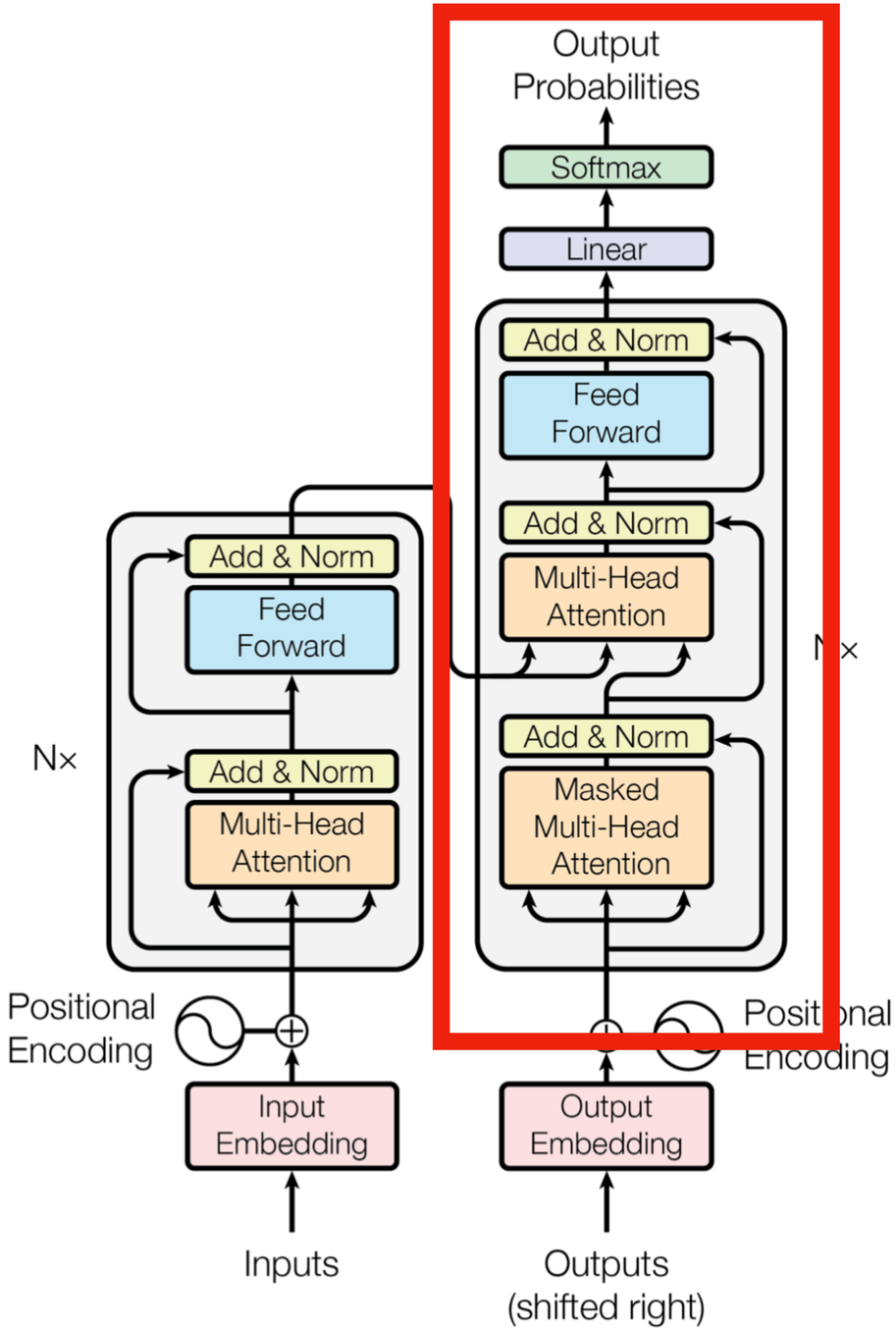
- For each token in the sequence, the model calculates query, key, and value vectors by applying learned linear transformations to the token embeddings
- The model computes the attention weights by measuring the similarity between the query vector of a token and the key vectors of all tokens in the sequence
- The attention weights obtained in the previous step are used to weight the value vectors of each token
- The resulting weighted sum is the transformed representation of the original token

	I	Love	Ice-Cream
I	.76	.89	.54
Love	.89	.46	.88
Ice-Cream	.54	.88	.54

	I	Love	Ice-Cream
I	.76	.89	.54
Love	.89	.46	.88
Ice-Cream	.54	.88	.54

Value

Encoder



Decoder

Figure 1: The Transformer - model architecture.

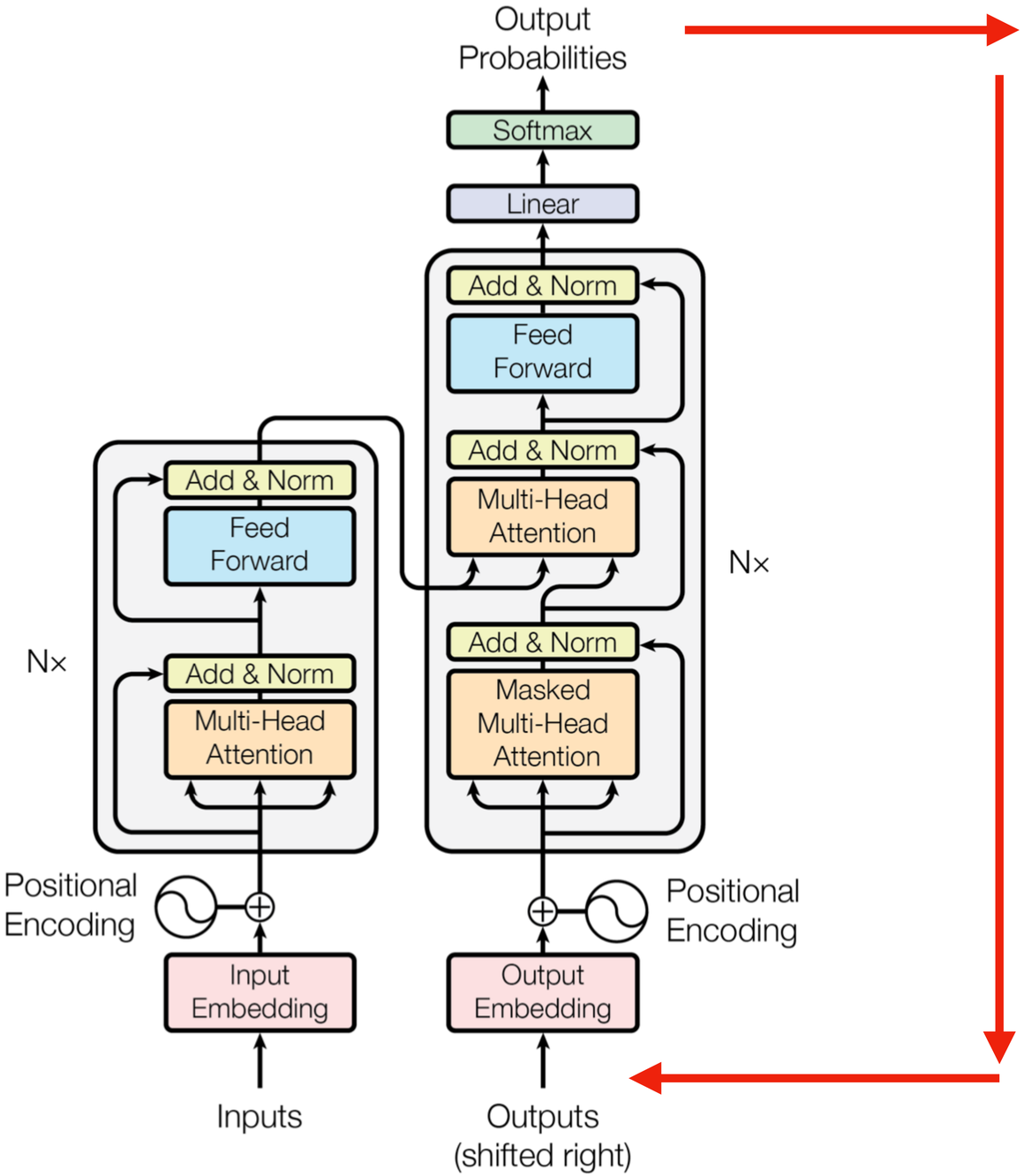


Figure 1: The Transformer - model architecture.

Why “Transformers”?

Why “Transformers”?

- model “transforms” the representation of each input token by attending to different positions in the sequence

Transformer Models for Social Science

- Reduction of labeled data, performance increase
- Pre-trained Language Models:
    - Pre-trained language models, such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) are used for text classification tasks.
    - These models learn contextual word representations from large amounts of unlabeled text data and can be fine-tuned for specific classification tasks.

Transformers for classification

- Gilardi, F., Alizadeh, M., & Kubli, M. (2023). Chatgpt outperforms crowd-workers for text-annotation tasks. arXiv preprint arXiv:2303.15056

- Huang, F., Kwak, H., & An, J. (2023). Is ChatGPT better than Human Annotators? Potential and Limitations of ChatGPT in Explaining Implicit Hate Speech. arXiv preprint arXiv:2302.07736

- Reiss, M. V. (2023). Testing the Reliability of ChatGPT for Text Annotation and Classification: A Cautionary Remark. arXiv preprint arXiv:2304.11085.

Transformers for “Feature Engineering”

- What features are the most important for (e.g.,) discriminating text types?
- Feed the model several example texts
- It distinguishes features that are important
- Use these features to classify/scale/etc., large quantities of data

Transformers for Multilingual TA

- Align corpora
- Machine Translation
- No need for translation?

Open vs. Closed source

- Ethics
- Performance vs. Reproducibility
- Use cases
    - Do you ***need*** it?

Open Source

- Huggingface:
    - https://huggingface.co/
    - Many pre-trained models and datasets
- spaCy:
    - https://spacy.io/
    - Ecosystem for NLP

Open Source

- Better in Python
- But:
    - spaCyR
    - https://github.com/chainsawriot/grafzahl
        - Huggingface wrapper

Your Thoughts?