L5: Deep Learning & Artificial Intelligence

Hobson Lane, UC San Diego
Instructor

UC San Diego EXTENSION

UC San Diego SCHOOL OF MEDICINE

Agenda

- Machine Learning Review
- Deep Learning
  - Layered linear regression
  - Forward and backward propagation
  - Making it nonlinear
  - What it's good for
  - Playground
- Artificial Intelligence
  - Intelligence Augmentation

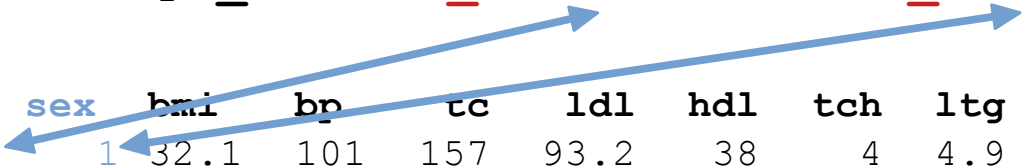
Linear Regression Review: Diabetes

$$y = c_0 + c_1 \cdot x_1 + c_2 \cdot x_2 + \ldots$$

	age	sex	bmi	bp	tc	ldl	hdl	tch	ltg	glu	severity
0	59	1	32.1	101	157	93.2	38	4	4.9	87	**151**
1	48	0	21.6	87	183	103.2	70	3	3.9	69	**75**
2	72	1	30.5	93	156	93.6	41	4	4.7	85	**141**
3	24	0	25.3	84	198	131.4	40	5	4.9	89	**206**
4	50	0	23.0	101	192	125.4	52	4	4.3	80	**135**
5	23	0	22.6	89	139	64.8	61	2	4.2	68	**97**
6	36	1	22.0	90	160	99.6	50	3	4	82	**138**
7	66	1	26.2	114	255	185.0	56	4.6	4.2	92	**63**
8	60	1	32.1	83	179	119.4	42	4	4.5	94	**110**
9	29	0	30.0	85	180	93.4	43	4	5.4	88	**310**
10	22	0	18.6	97	114	57.6	46	2	4	83	**101**
11	56	1	28.0	85	184	144.8	32	6	3.6	77	**69**
12	53	0	23.7	92	186	109.2	62	3	4.3	81	**179**

LinearRegression().fit(X,y).coef_

```
y = intercept_ + coef_[0]*x[0] + coef_[1]*x[1] + …
```

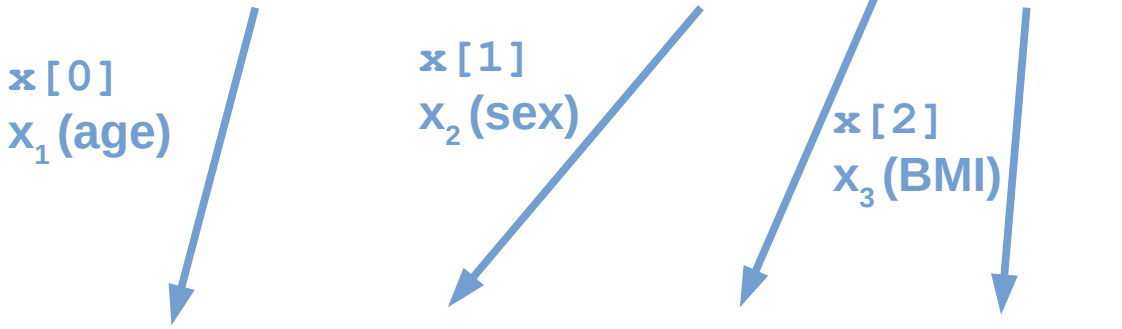


	age	sex	bmi	bp	tc	ldl	hdl	tch	ltg	glu	severity
0	59	1	32.1	101	157	93.2	38	4	4.9	87	151
1	48	0	21.6	87	183	103.2	70	3	3.9	69	75
2	72	1	30.5	93	156	93.6	41	4	4.7	85	141
3	24	0	25.3	84	198	131.4	40	5	4.9	89	206
4	50	0	23.0	101	192	125.4	52	4	4.3	80	135
5	23	0	22.6	89	139	64.8	61	2	4.2	68	97
6	36	1	22.0	90	160	99.6	50	3	4	82	138
7	66	1	26.2	114	255	185.0	56	4.6	4.2	92	63
8	60	1	32.1	83	179	119.4	42	4	4.5	94	110
9	29	0	30.0	85	180	93.4	43	4	5.4	88	310
10	22	0	18.6	97	114	57.6	46	2	4	83	101
11	56	1	28.0	85	184	144.8	32	6	3.6	77	69
12	53	0	23.7	92	186	109.2	62	3	4.3	81	179

Data Flow

age	sex	bmi	bp	tc	…
59	2	32.1	101	157	…

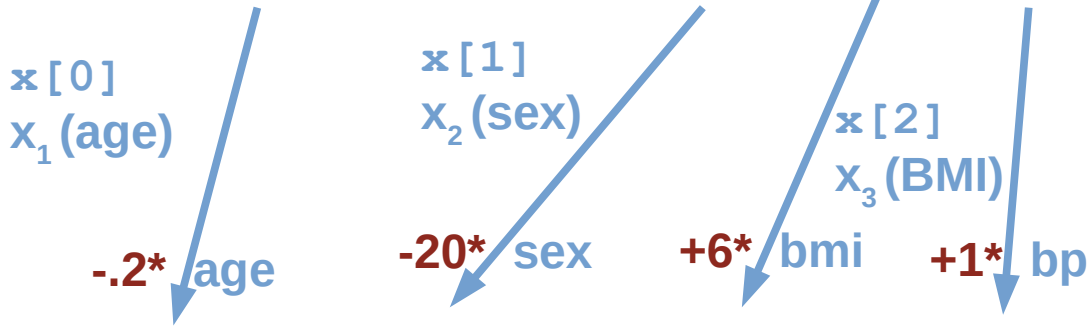
```
y = intercept_ + coef_[0]*x[0] + coef_[1]*x[1] + …    x[3]
```



Multiplication

age	sex	bmi	bp	tc	…
59	1	32.1	101	157	…

```
y = intercept_ + coef_[0]*x[0] + coef_[1]*x[1] + …    x[3]
```



Coefficients as Slopes

```
y = intercept_ + coef_[0]*x[0] + coef_[1]*x[1] + ...
```

x[3]

x[0]
x_1 (age)

x[1]
x_2 (sex)

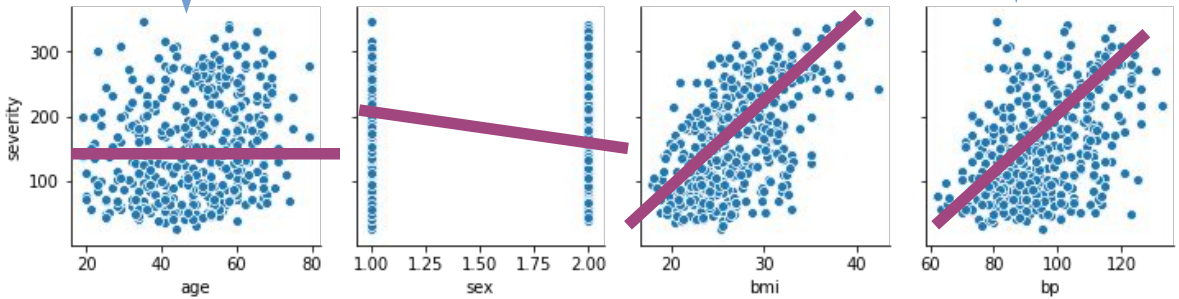
x[2]
x_3 (BMI)

-.2* age

-20* sex

+6* bmi

+1* bp



Combine the Nudges

```
y = intercept_ + coef_[0]*x[0] + coef_[1]*x[1] + ...
```

x[3]

x[0]
x_1 (age)

x[1]
x_2 (sex)

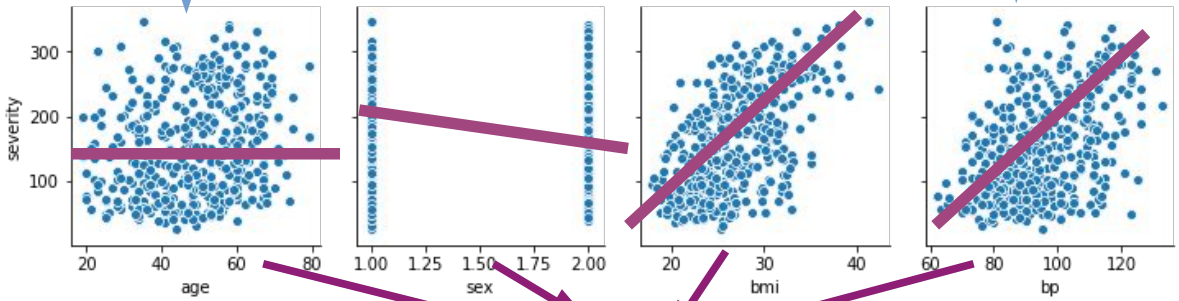
x[2]
x_3 (BMI)

-.2* age

-20* sex

+6* bmi

+1* bp

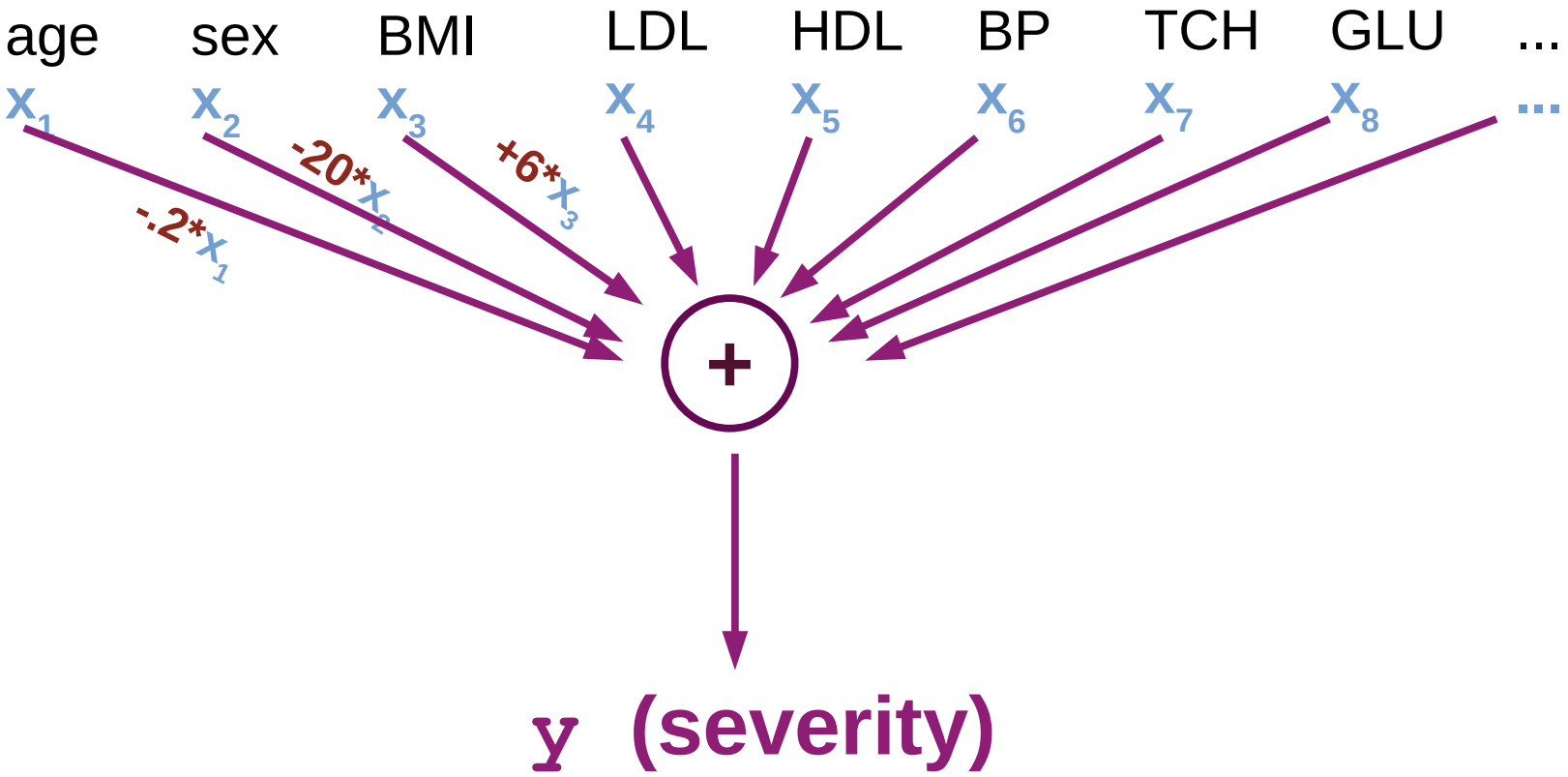


150*1.0+

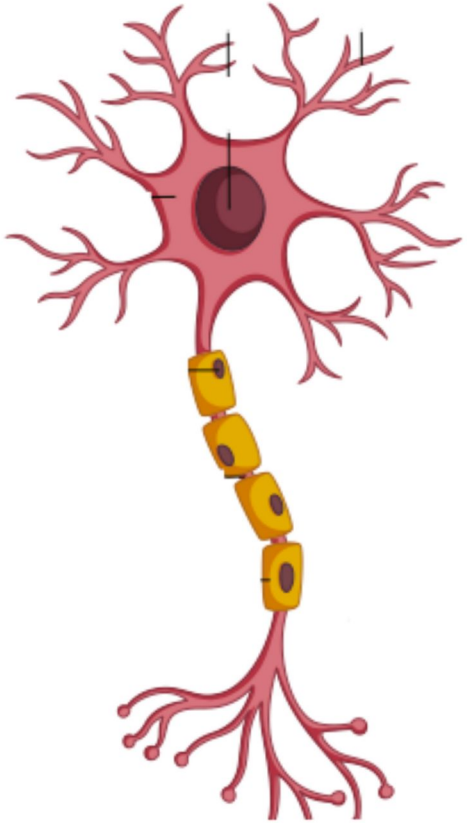
+

y (severity)

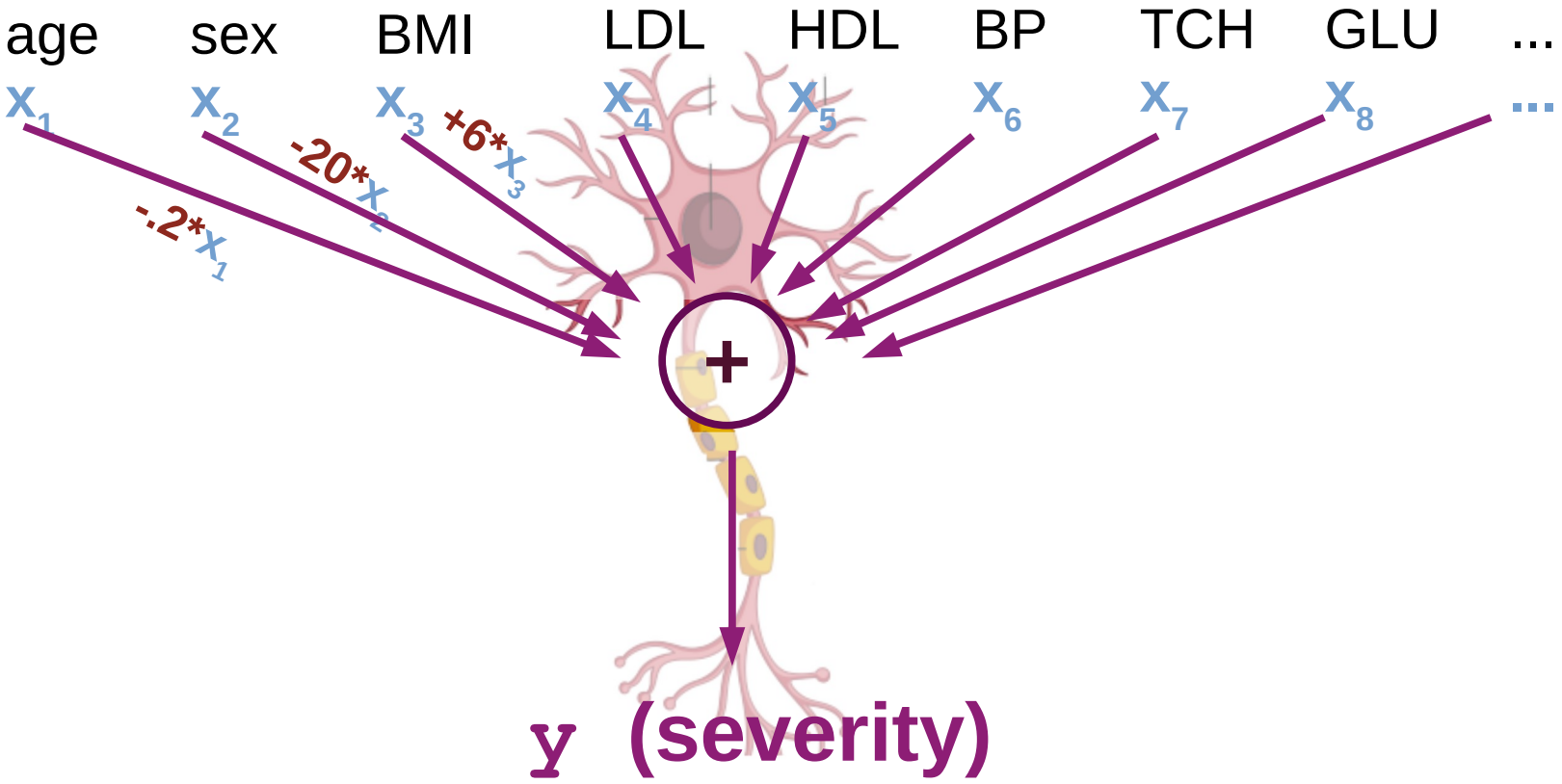
Network Diagram



Neuron

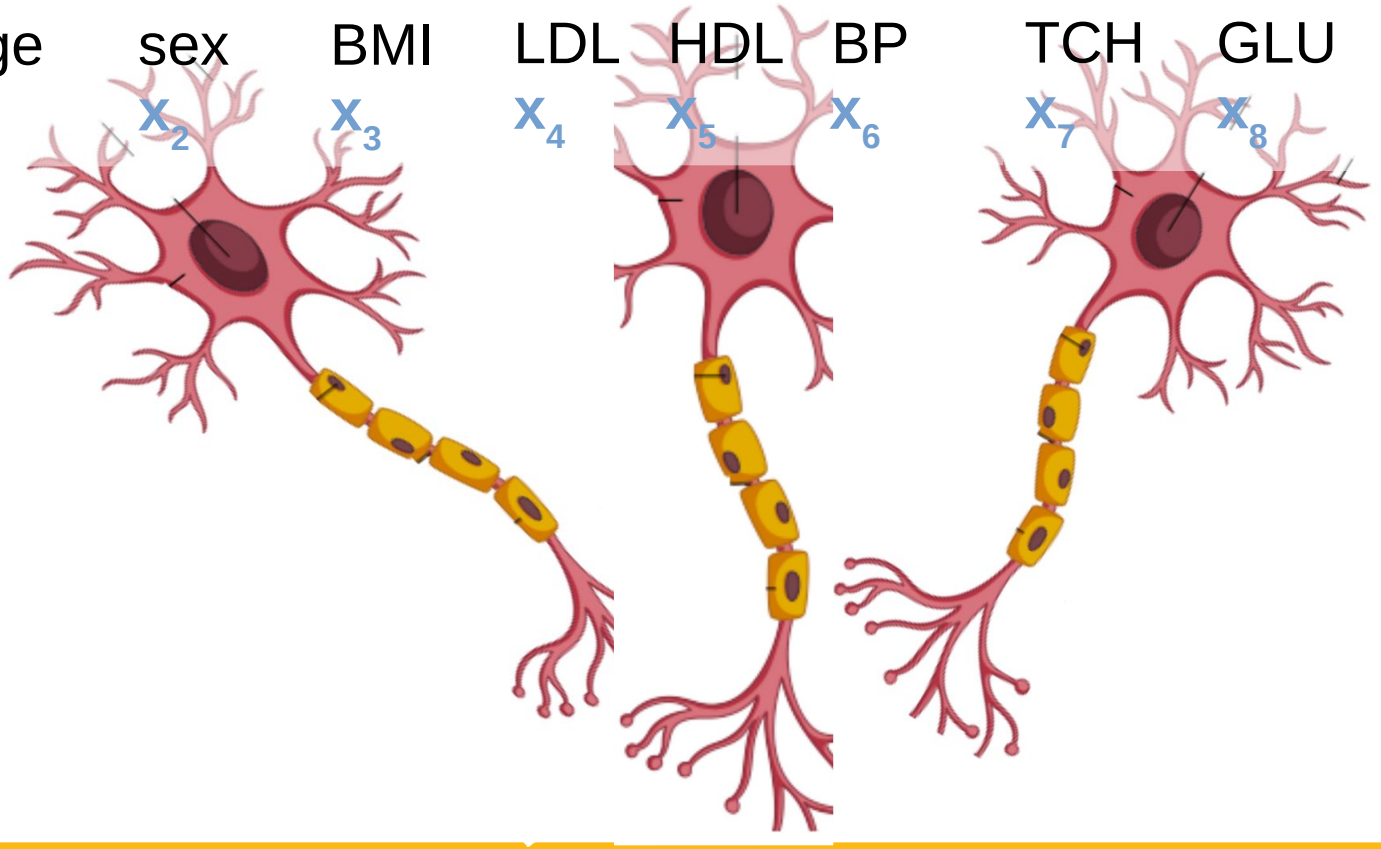


Perceptron = Neuron Simulation

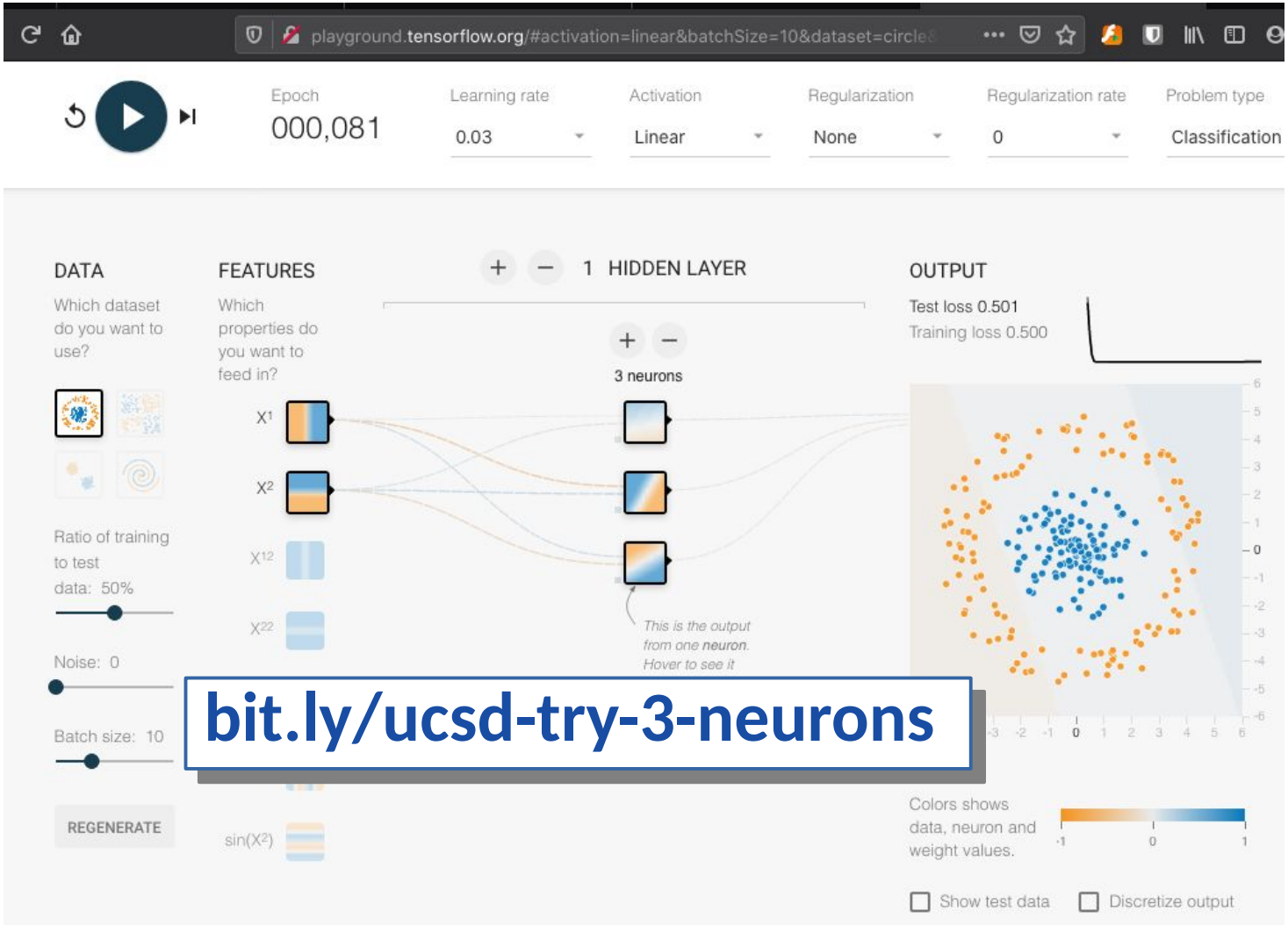


More neurons = smarter?

age x_1 sex x_2 BMI x_3 LDL x_4 HDL x_5 BP x_6 TCH x_7 GLU x_8 ... ...

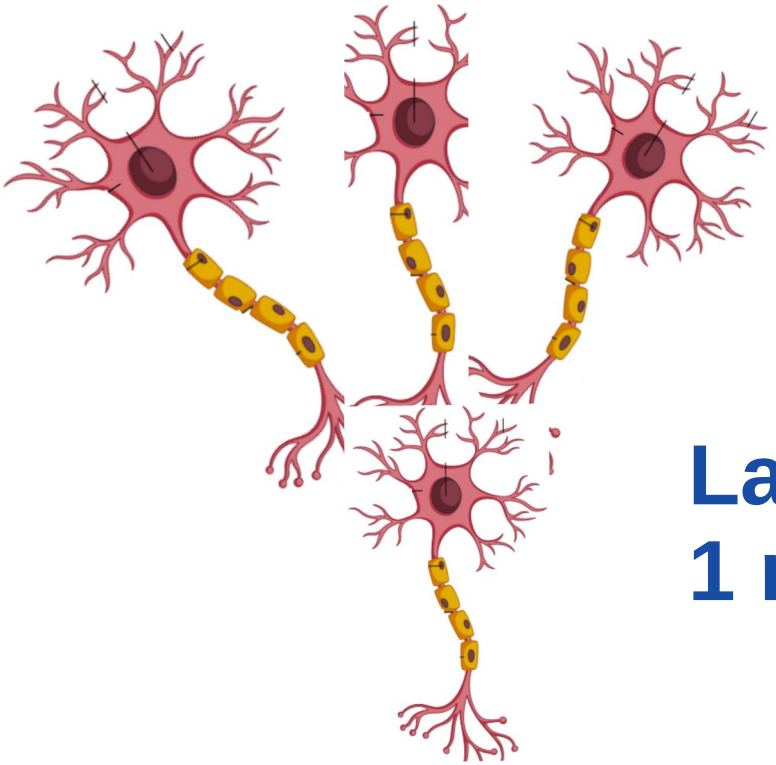


Try it!



bit.ly/ucsd-try-3-neurons

More layers = smarter?

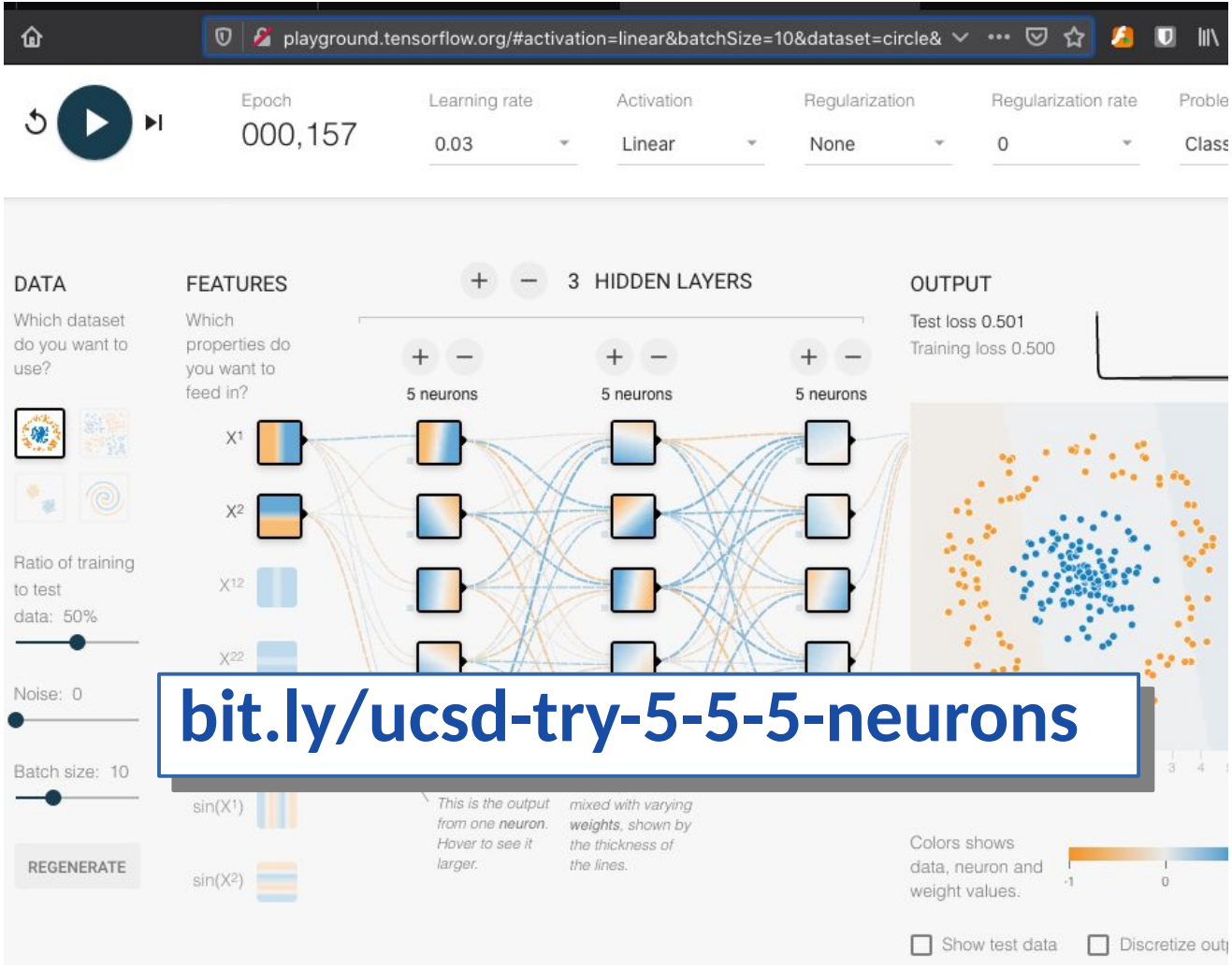


Layer 1
3 neurons

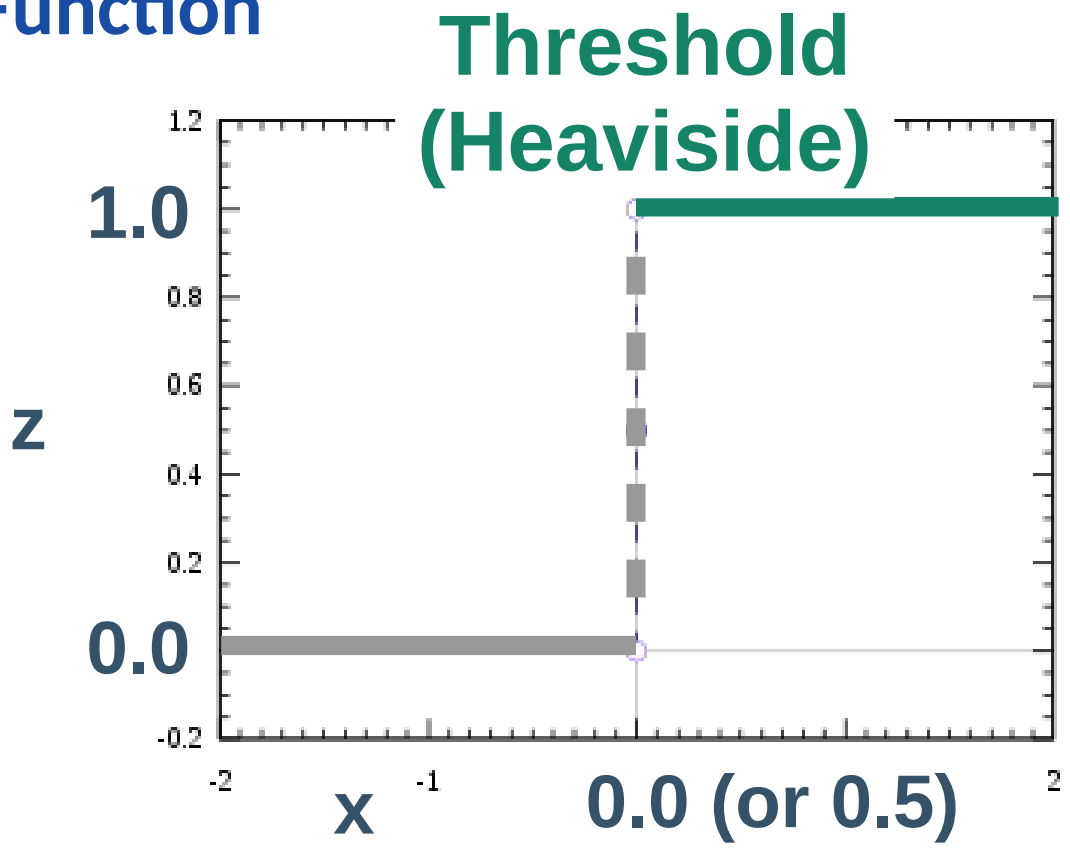
Layer 2
1 neuron

More

Power?

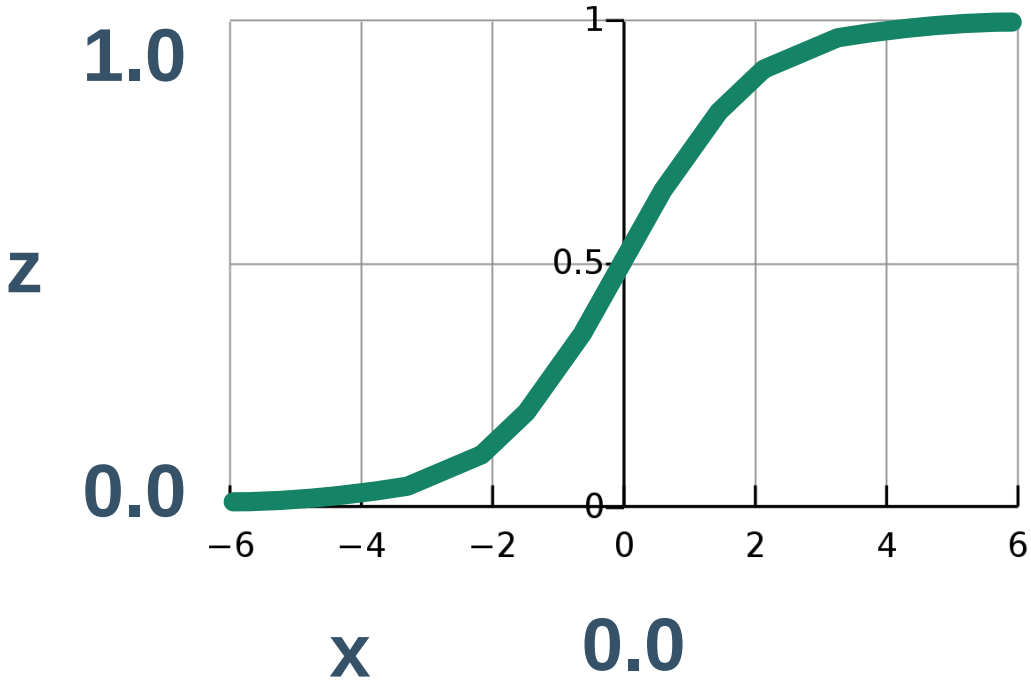

playground.tensorflow.org/#activation=linear&batchSize=10&dataset=circle&
Epoch
000,157
Learning rate
0.03
Activation
Linear
Regularization
None
Regularization rate
0
DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE
FEATURES
Which properties do you want to feed in?
3 HIDDEN LAYERS
5 neurons
5 neurons
5 neurons
OUTPUT
Test loss 0.501
Training loss 0.500
bit.ly/ucsd-try-5-5-5-neurons
This is the output from one neuron. Hover to see it larger.
mixed with varying weights, shown by the thickness of the lines.
Colors shows data, neuron and weight values.
Show test data

Activation Function

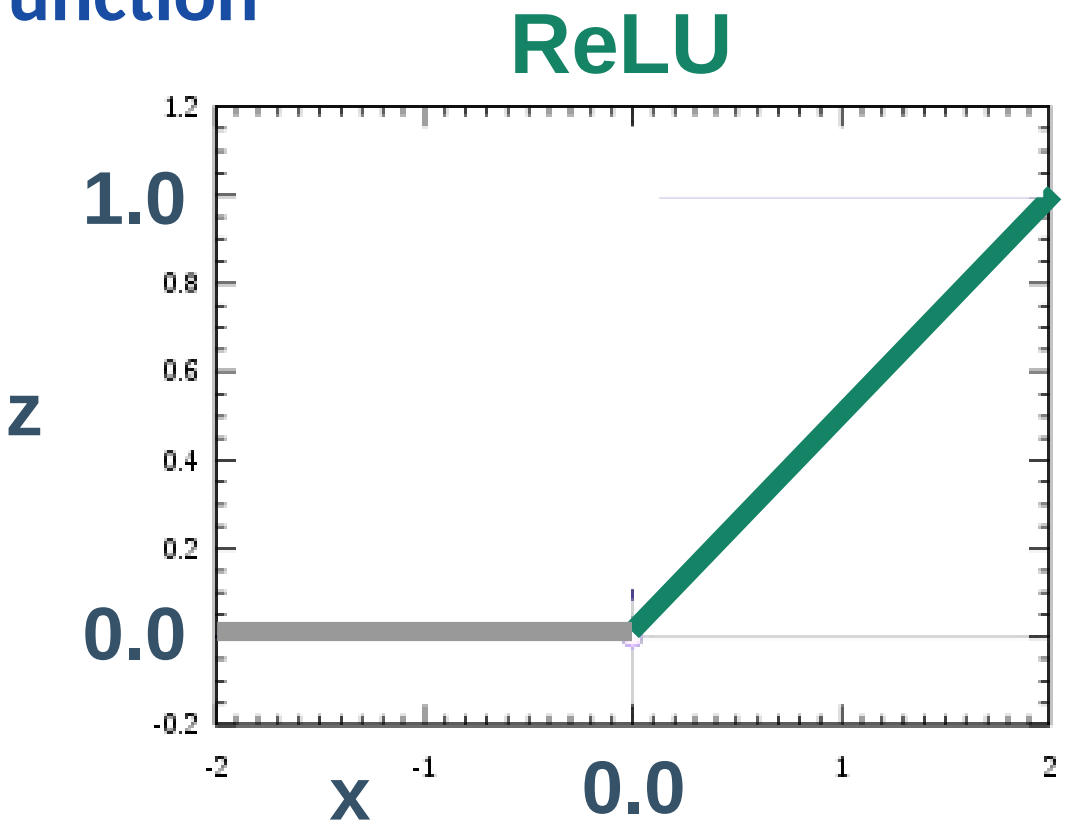


Activation Function

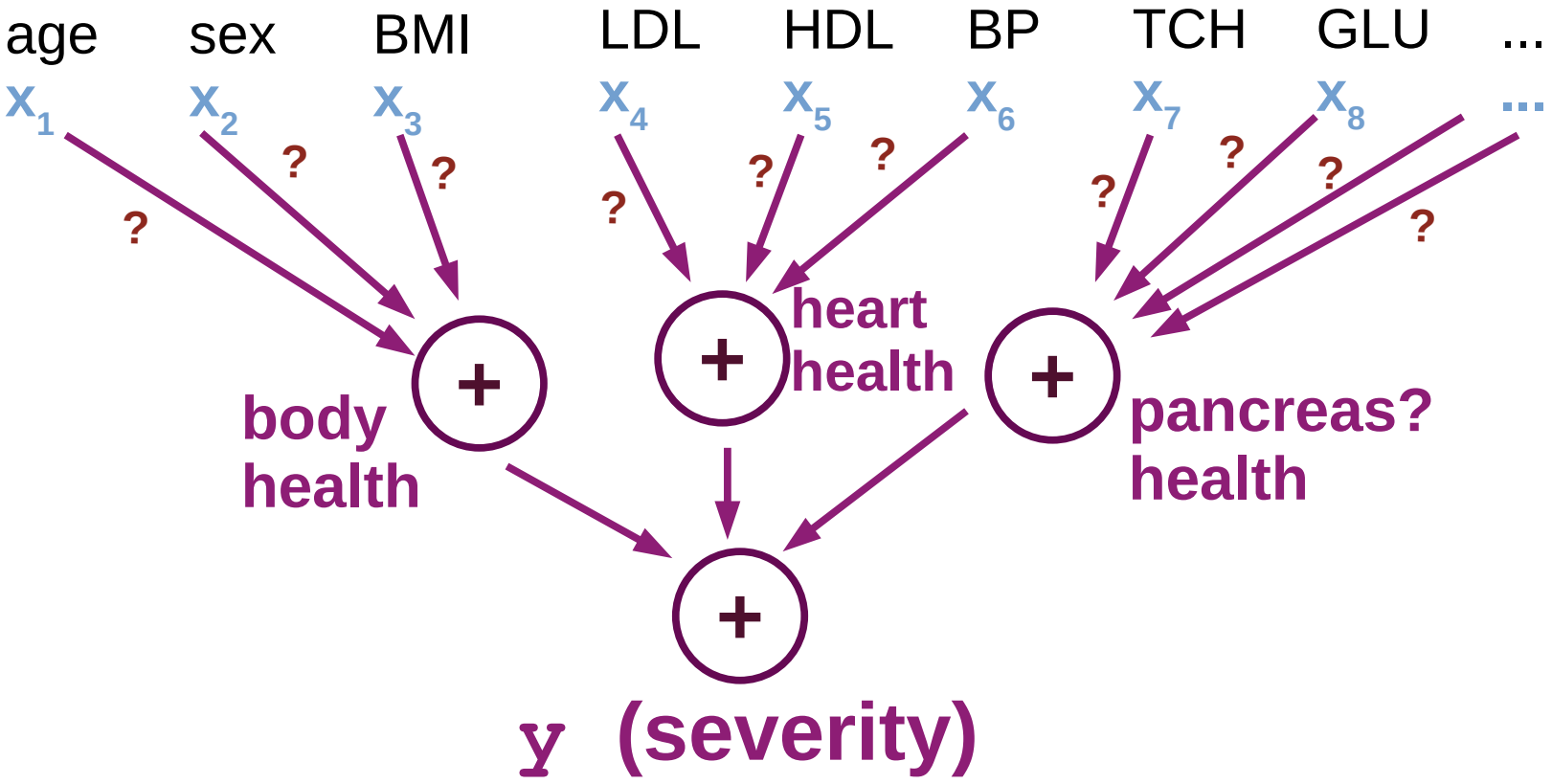
Sigmoid



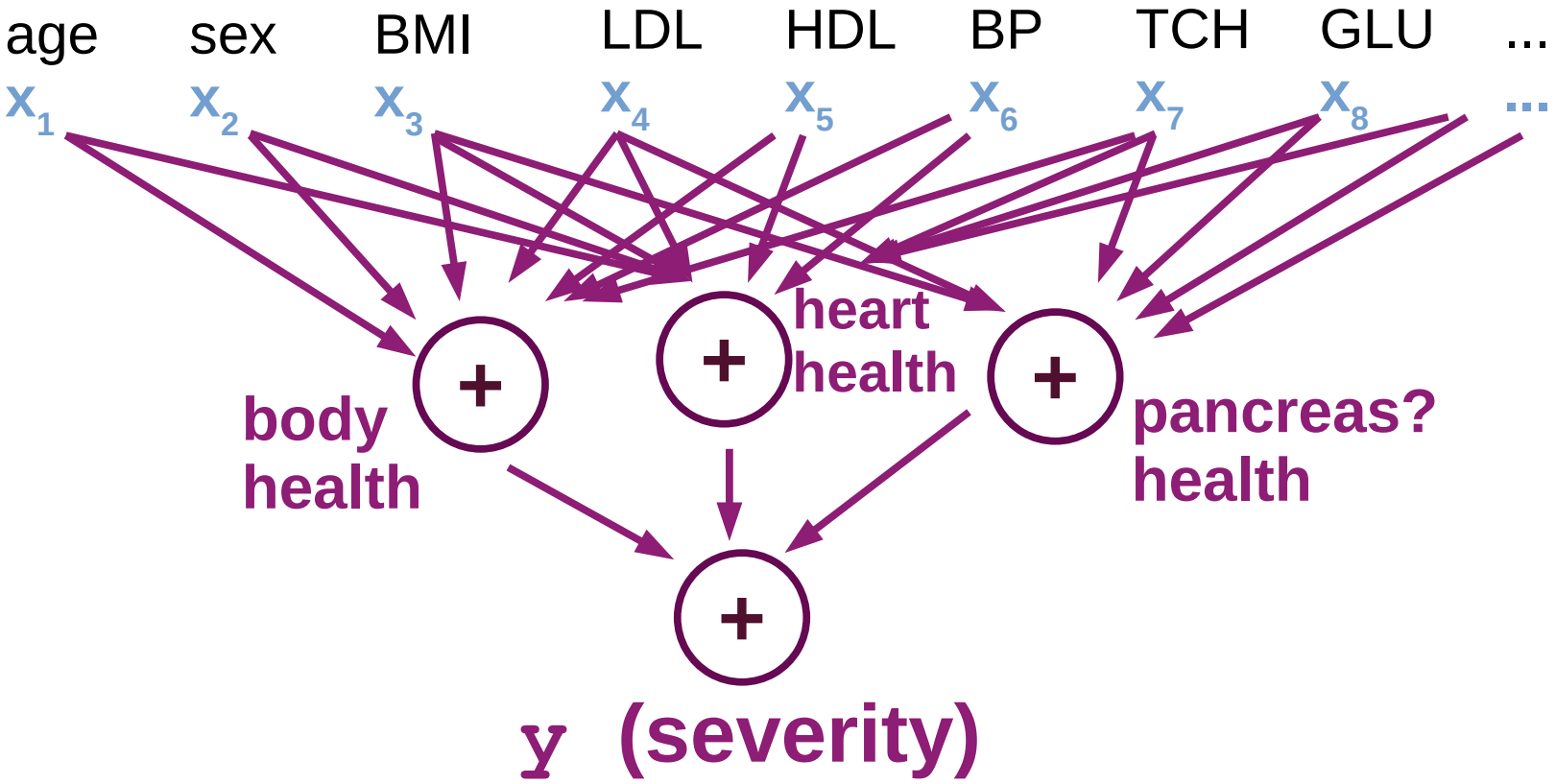
Activation Function



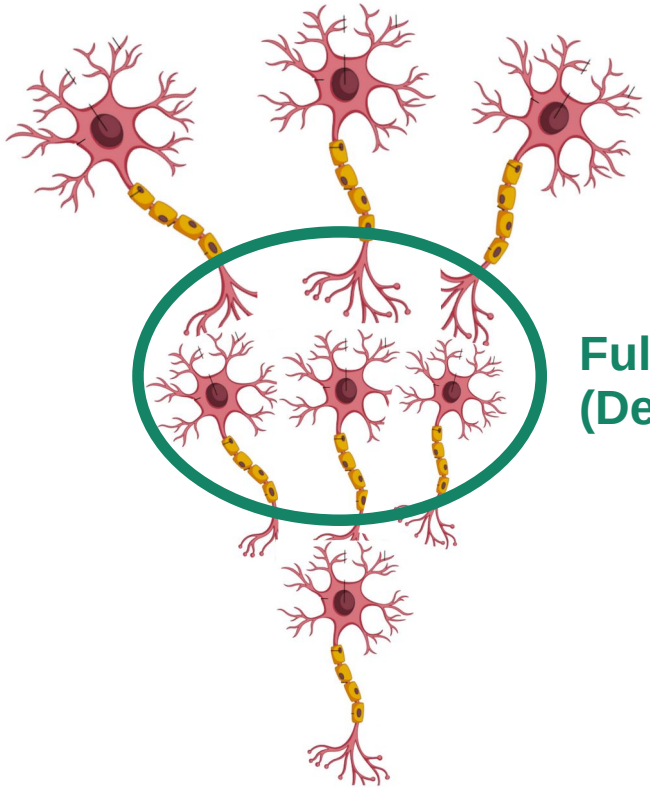
Grouping Features Together



Dense (Fully Connect) Network

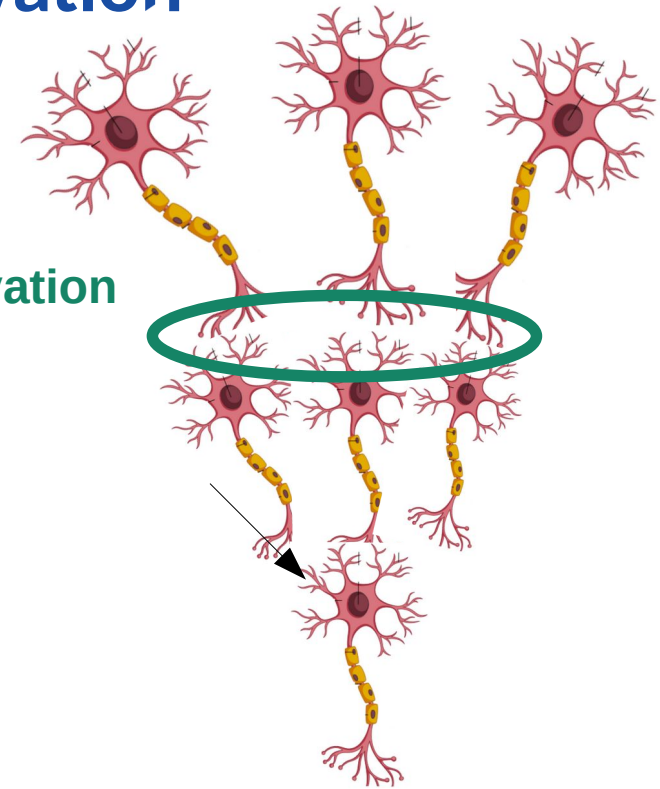


Your Network is Dense

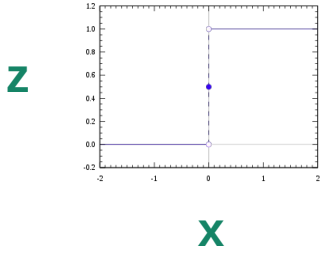


Fully Connected
(Dense)

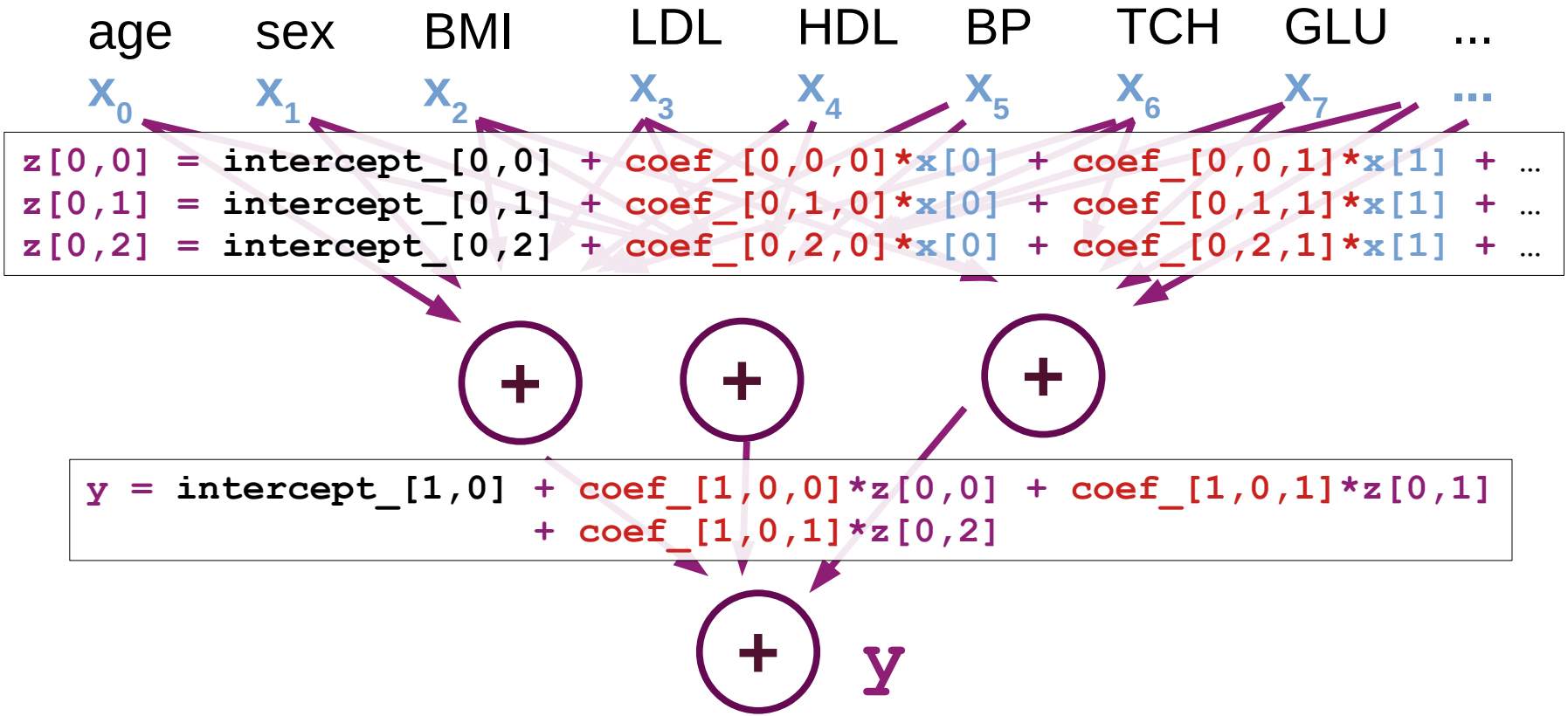
Synapse Activation



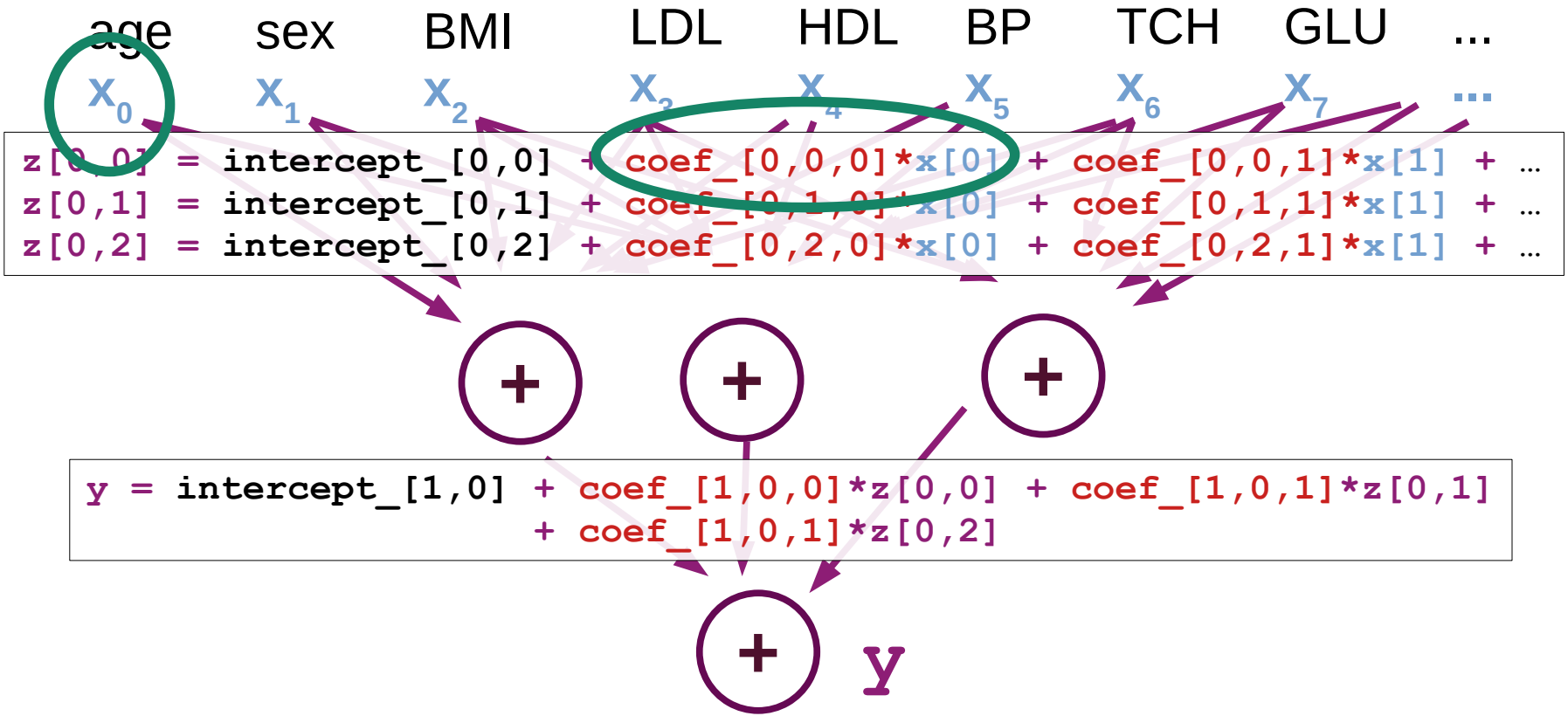
Synapse Activation Function



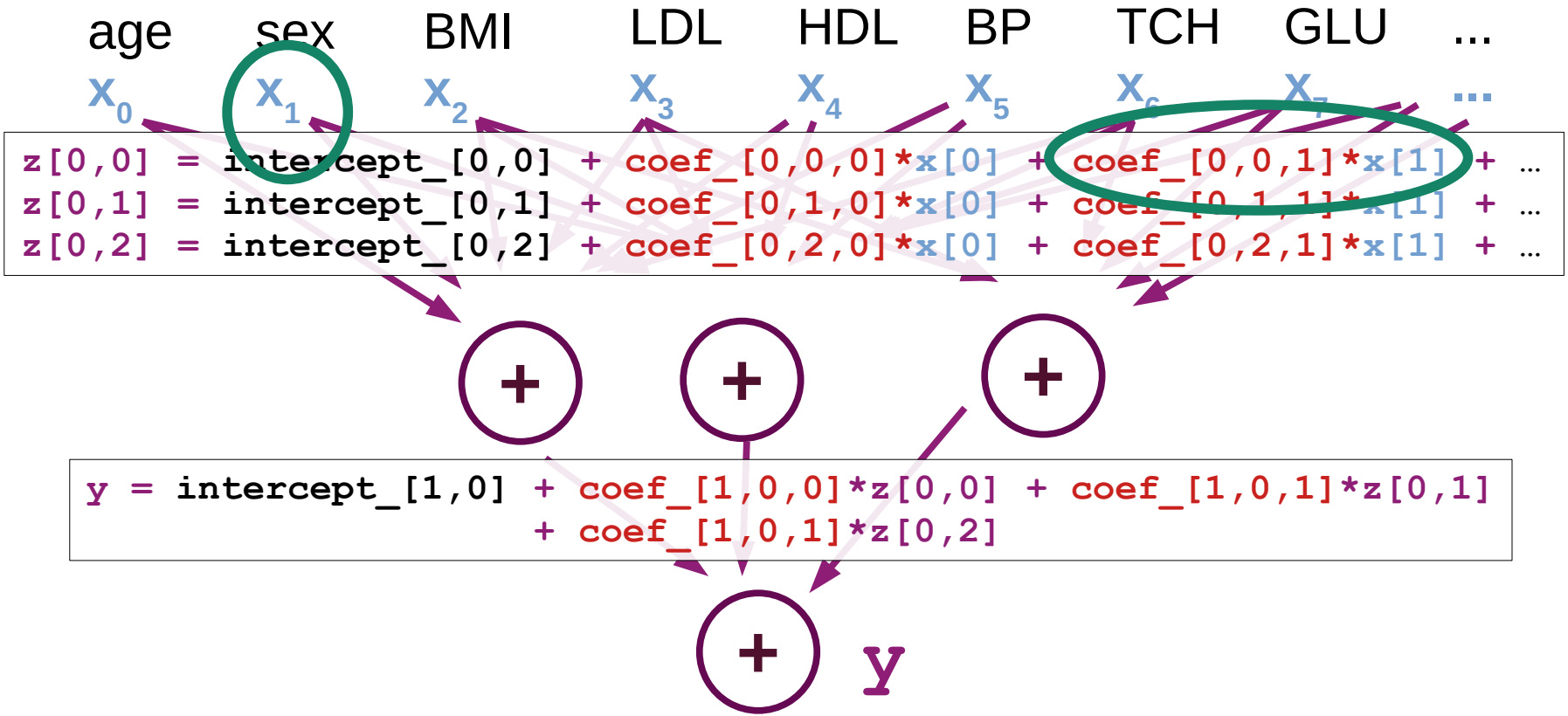
Dot Products in Python



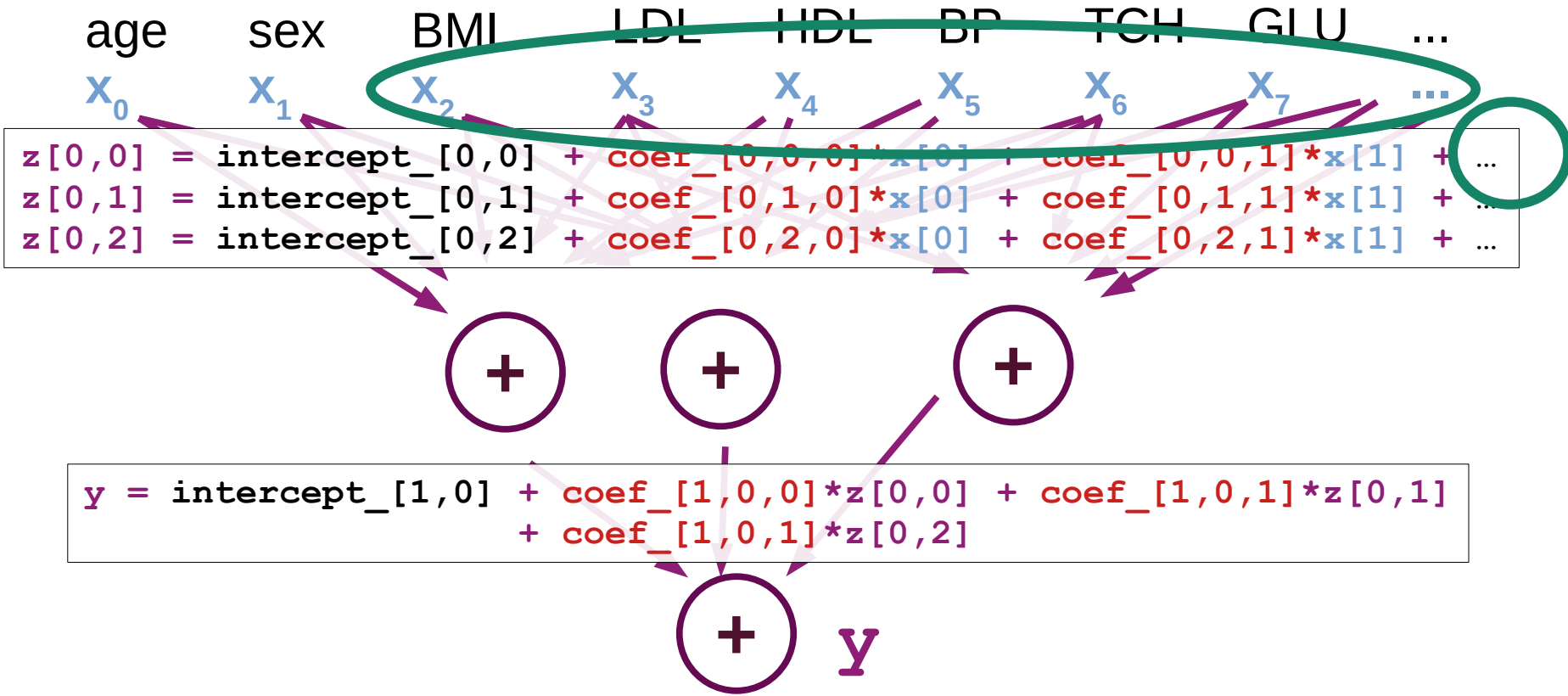
Input 1 (age)


age sex BMI LDL HDL BP TCH GLU ...
x0 x1 x2 x3 x4 x5 x6 x7 ...
z[0,0] = intercept_[0,0] + coef_[0,0,0]*x[0] + coef_[0,0,1]*x[1] + ...
z[0,1] = intercept_[0,1] + coef_[0,1,0]*x[0] + coef_[0,1,1]*x[1] + ...
z[0,2] = intercept_[0,2] + coef_[0,2,0]*x[0] + coef_[0,2,1]*x[1] + ...
+
+
+
y = intercept_[1,0] + coef_[1,0,0]*z[0,0] + coef_[1,0,1]*z[0,1]
+ coef_[1,0,1]*z[0,2]
+
y

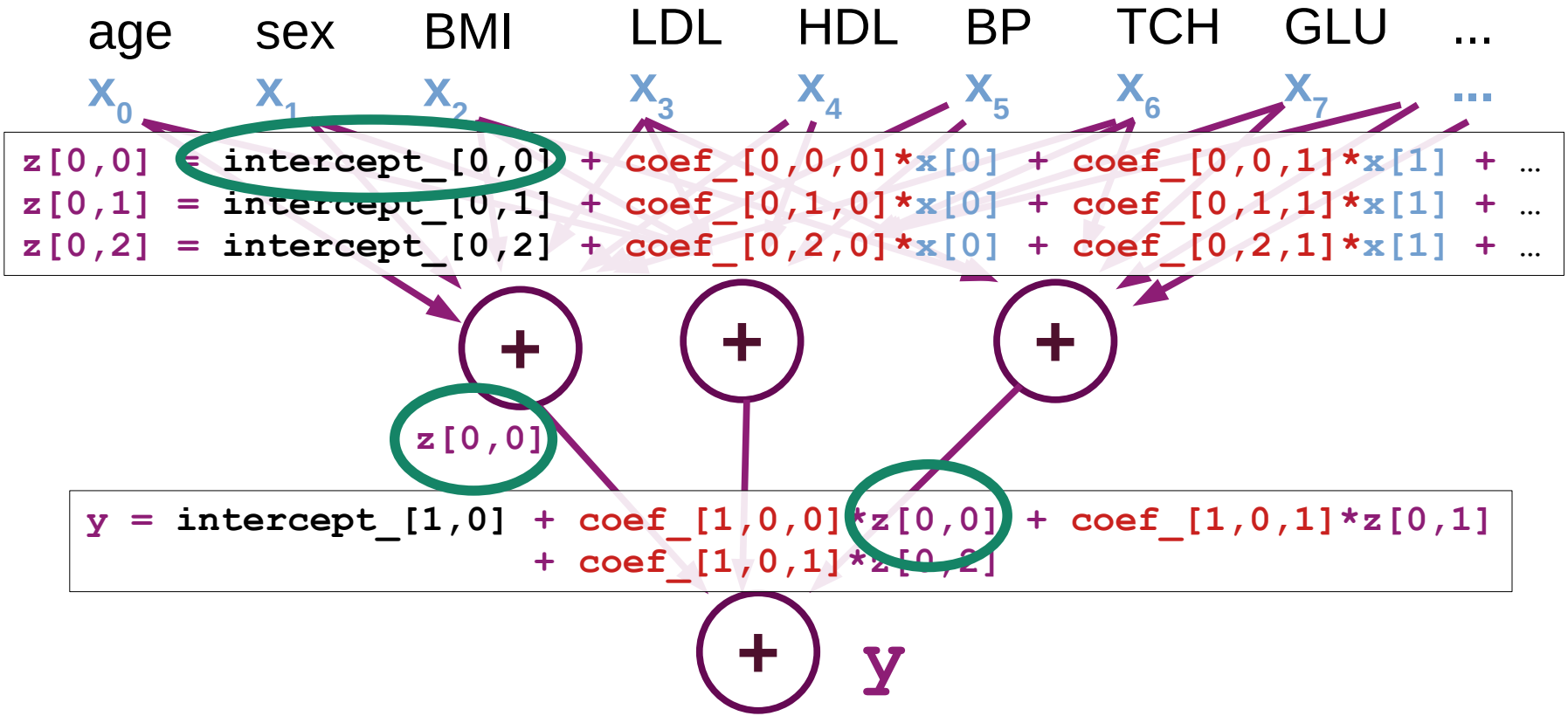
Input 2 (sex)


age
sex
BMI
LDL
HDL
BP
TCH
GLU
...
x_0
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
z[0,0] = intercept_[0,0] + coef_[0,0,0]*x[0] + coef_[0,0,1]*x[1] + ...
z[0,1] = intercept_[0,1] + coef_[0,1,0]*x[0] + coef_[0,1,1]*x[1] + ...
z[0,2] = intercept_[0,2] + coef_[0,2,0]*x[0] + coef_[0,2,1]*x[1] + ...
+
+
+
y = intercept_[1,0] + coef_[1,0,0]*z[0,0] + coef_[1,0,1]*z[0,1]
+ coef_[1,0,1]*z[0,2]
+
y

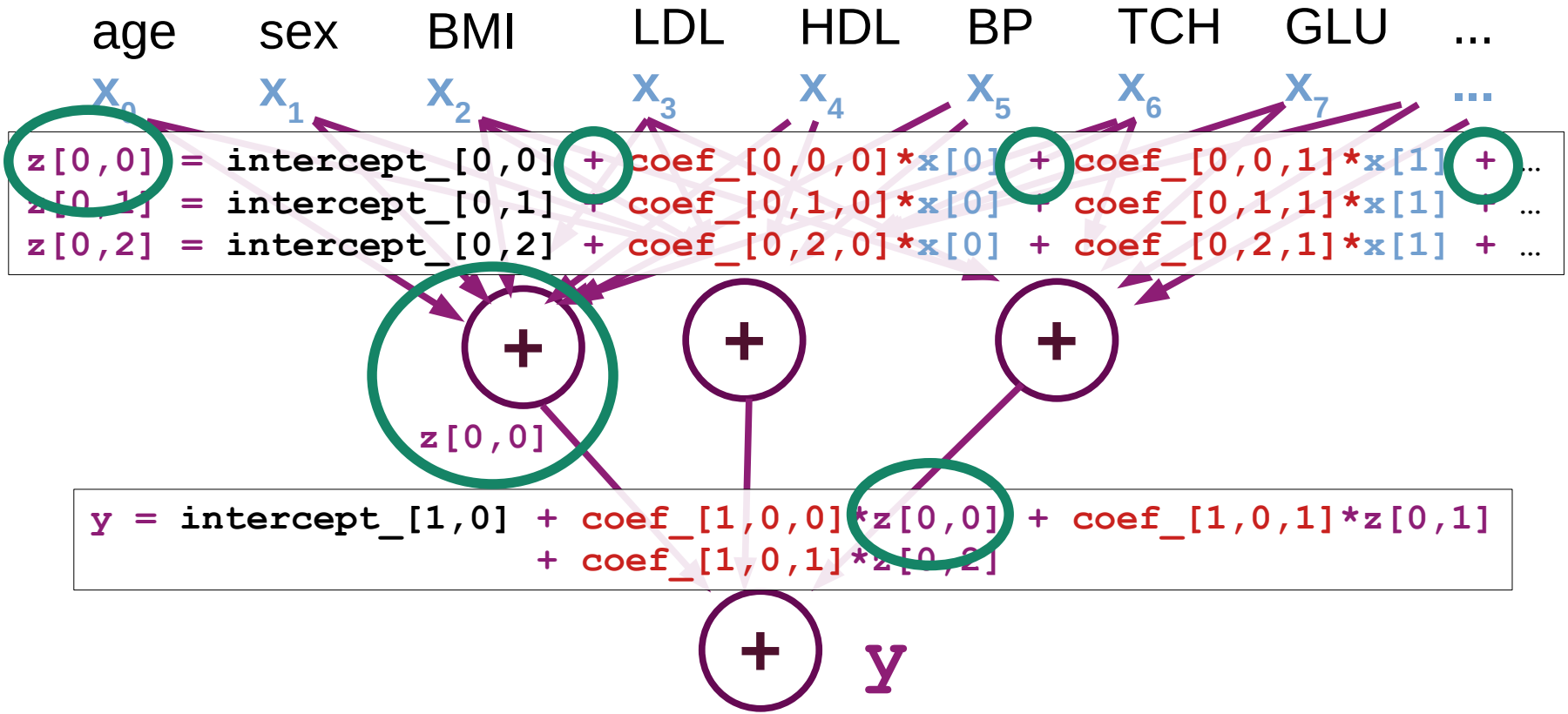
Inputs 3-10


age
sex
BMI
LDL
HDL
BP
TCH
GLU
...
x_0
x_1
x_2
x_3
x_4
x_5
x_6
x_7
...
z[0,0] = intercept_[0,0] + coef_[0,0,0]*x[0] + coef_[0,0,1]*x[1] + ...
z[0,1] = intercept_[0,1] + coef_[0,1,0]*x[0] + coef_[0,1,1]*x[1] + ...
z[0,2] = intercept_[0,2] + coef_[0,2,0]*x[0] + coef_[0,2,1]*x[1] + ...
+
+
+
y = intercept_[1,0] + coef_[1,0,0]*z[0,0] + coef_[1,0,1]*z[0,1]
+ coef_[1,0,1]*z[0,2]
+
y

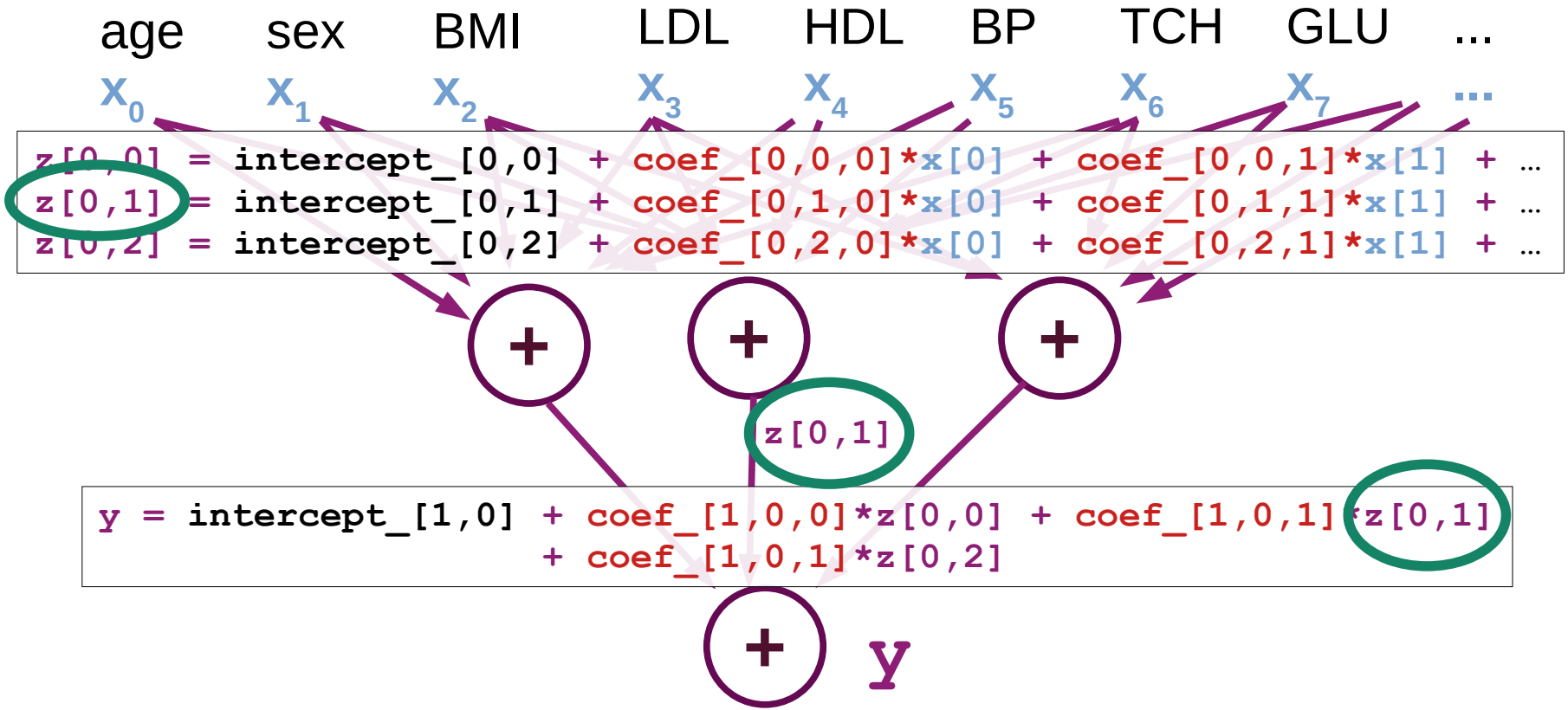
Layer 1, Intercept



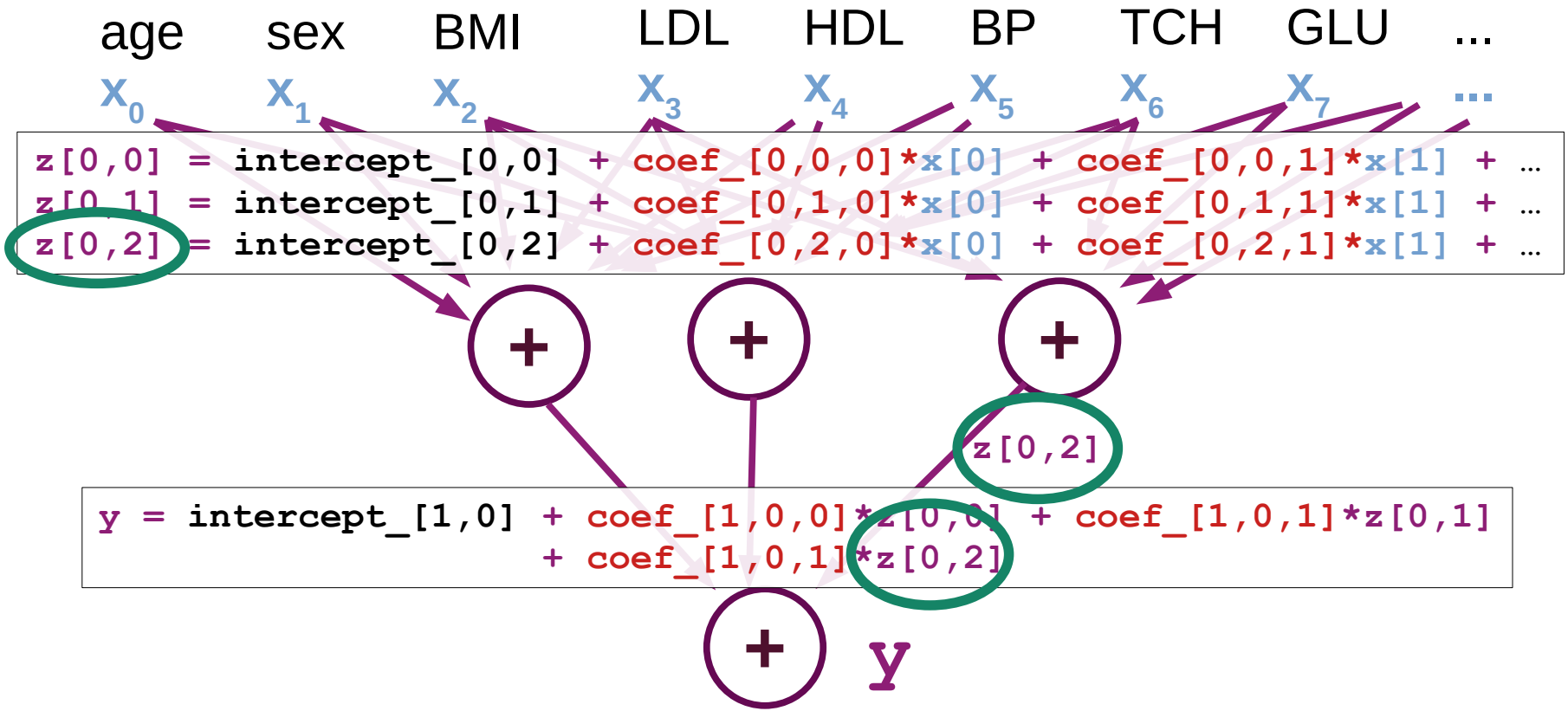
Layer 1, Neuron 1



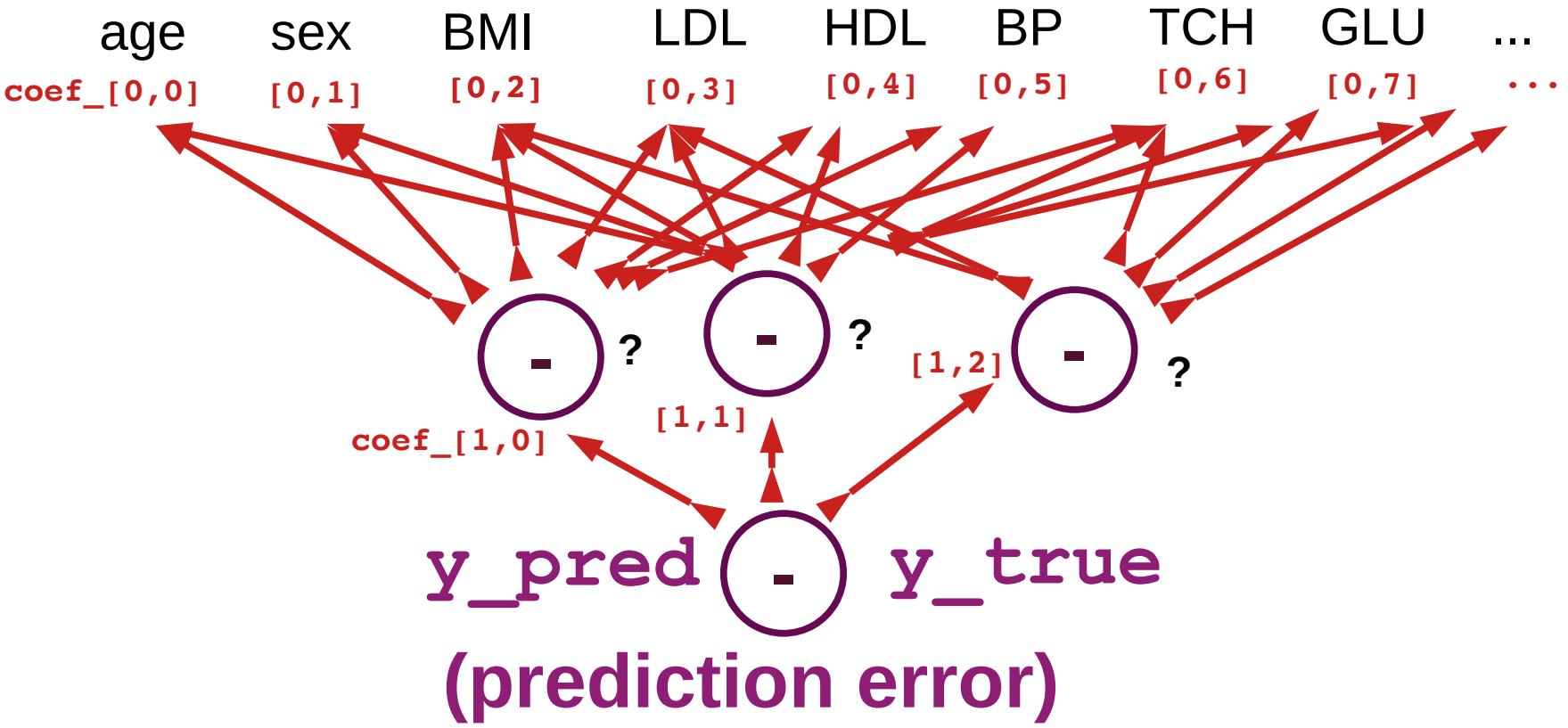
Layer 1, Neuron 2



Layer 1, Neuron 3



Backpropagation



Backpropagation: Prediction Error Gradients