

MYSTERY  
MACHINE  
LEARNING



redislabs  
HOME OF REDIS



**redis**labs  
HOME OF REDIS

# Guy Royse

## Developer Advocate

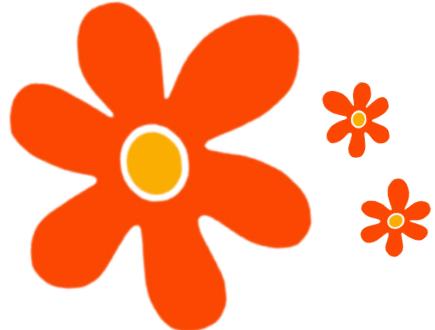
## Redis Labs

 @guyroyse

 [github.com/guyroyse](https://github.com/guyroyse)

 [guy.dev](https://guy.dev)

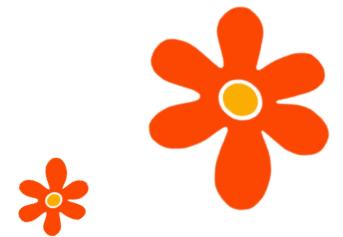
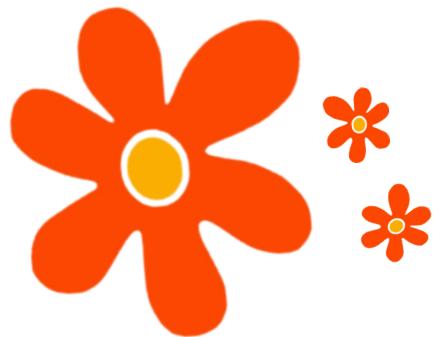
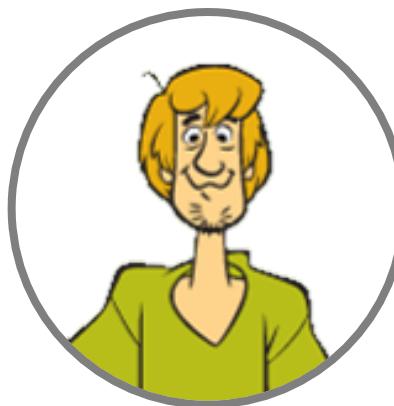
# IANADS



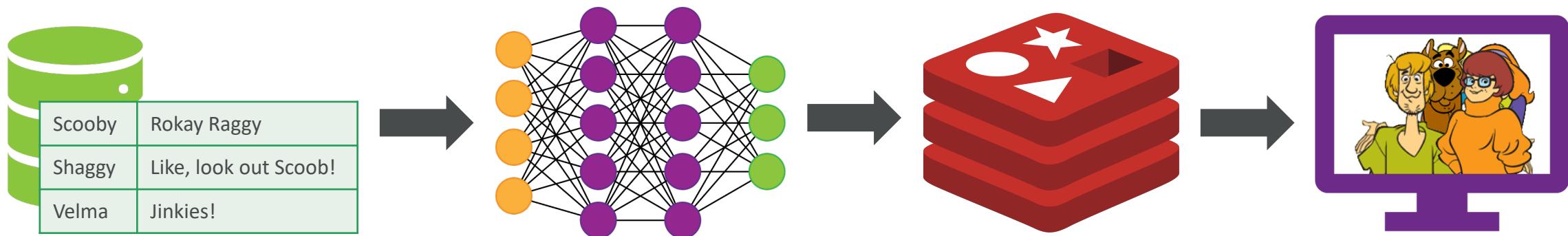
# SCOOBY-DOO, WHERE ARE YOU!

© COPYRIGHT 1969 HANNA-BARBERA PRODUCTIONS, INC.

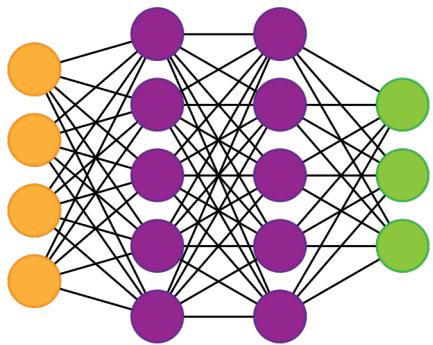
# WE'VE GOT A MYSTERY TO SOLVE



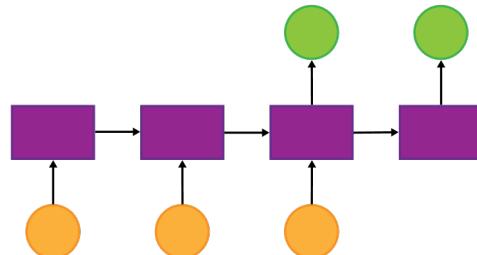
# WE'VE GOT SOME WORK TO DO NOW



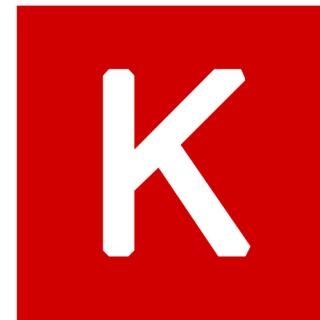
# WE NEED SOME HELP FROM YOU NOW



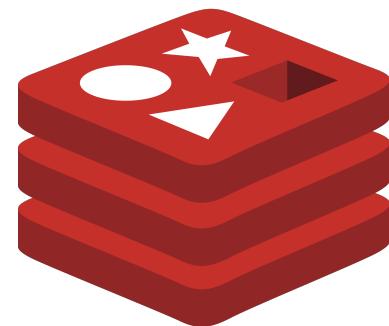
Neural Networks



Recurrent  
Neural Networks

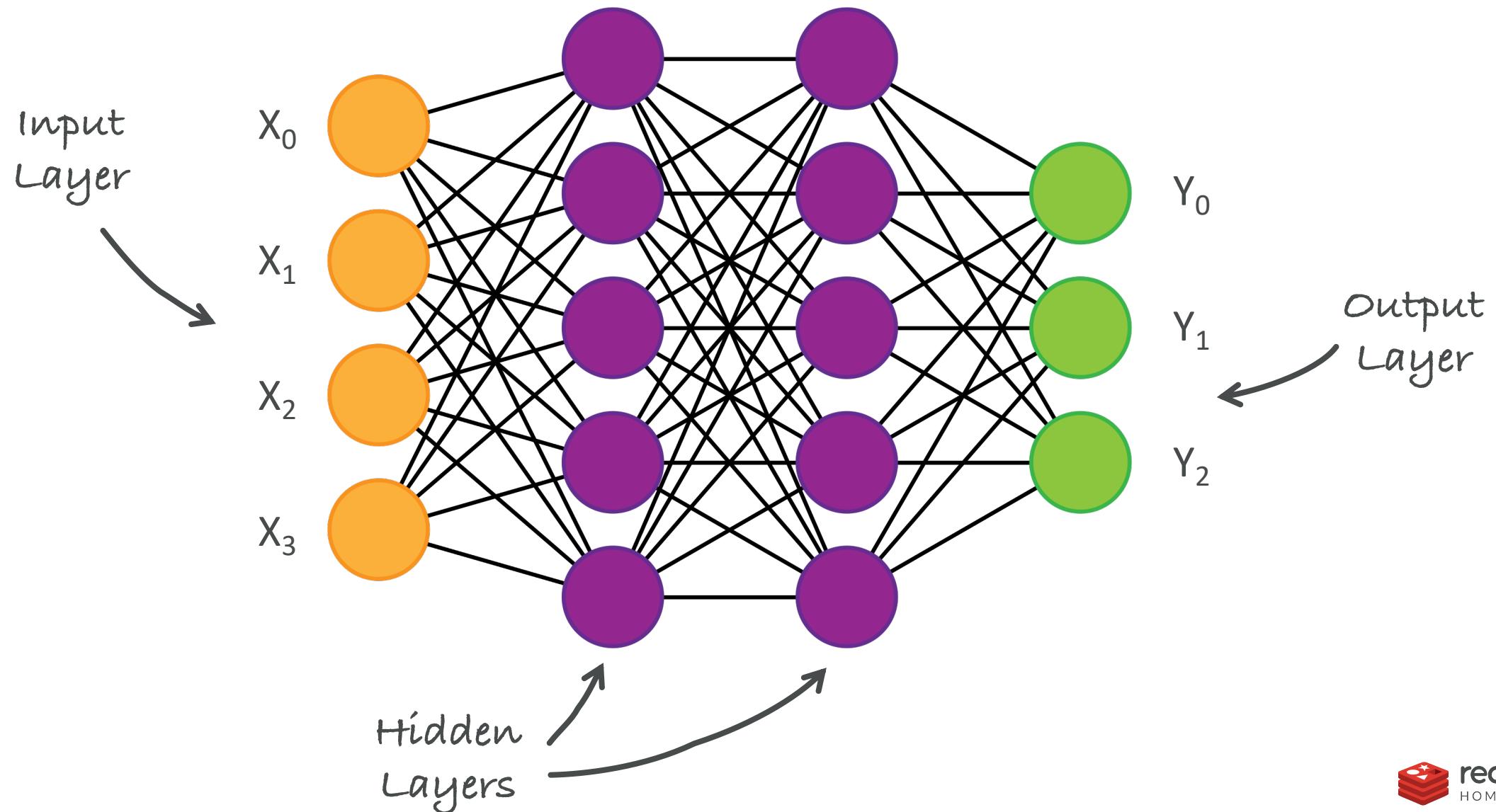


Keras

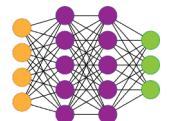
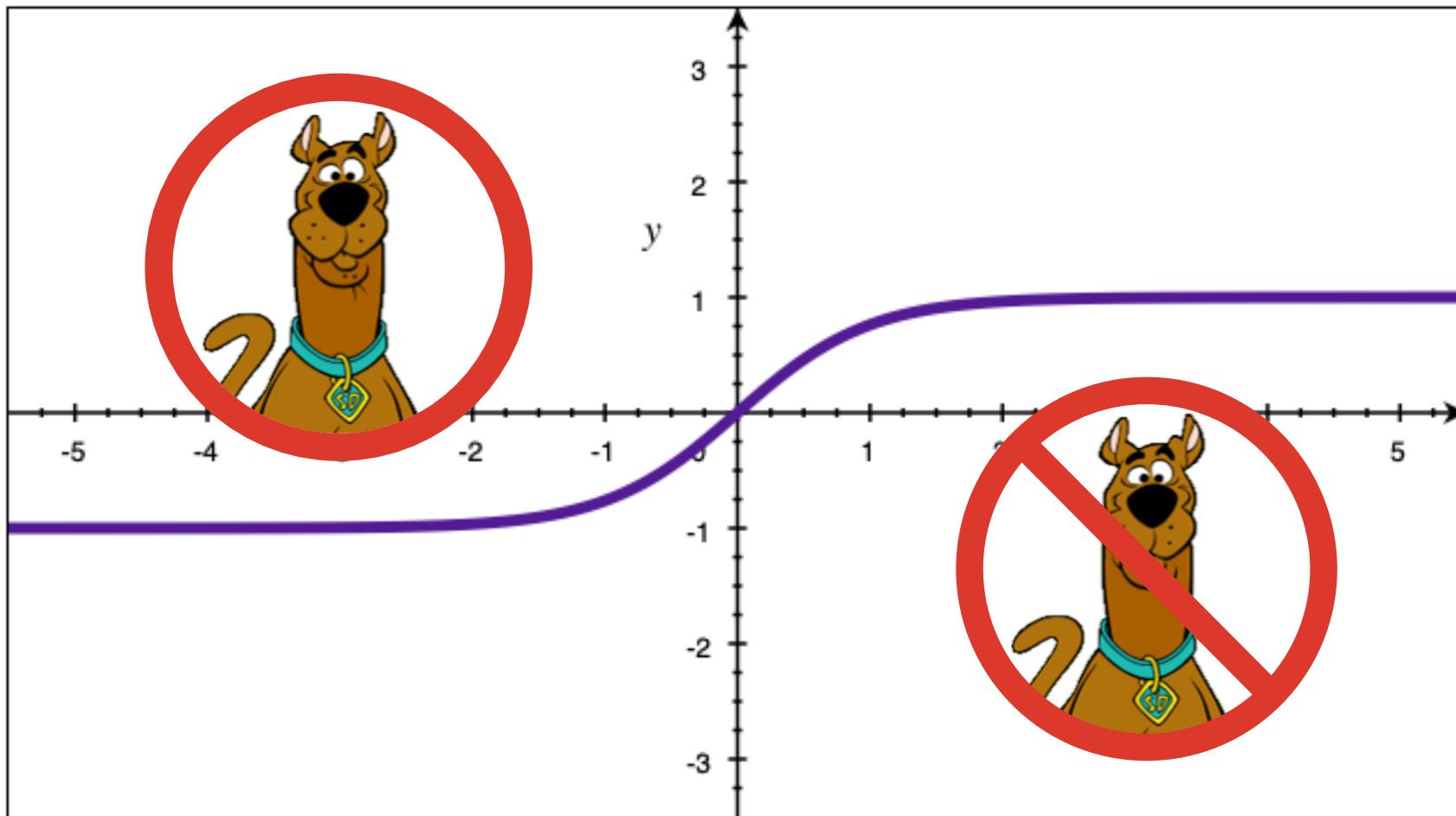


Redis

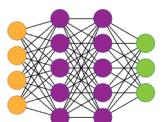
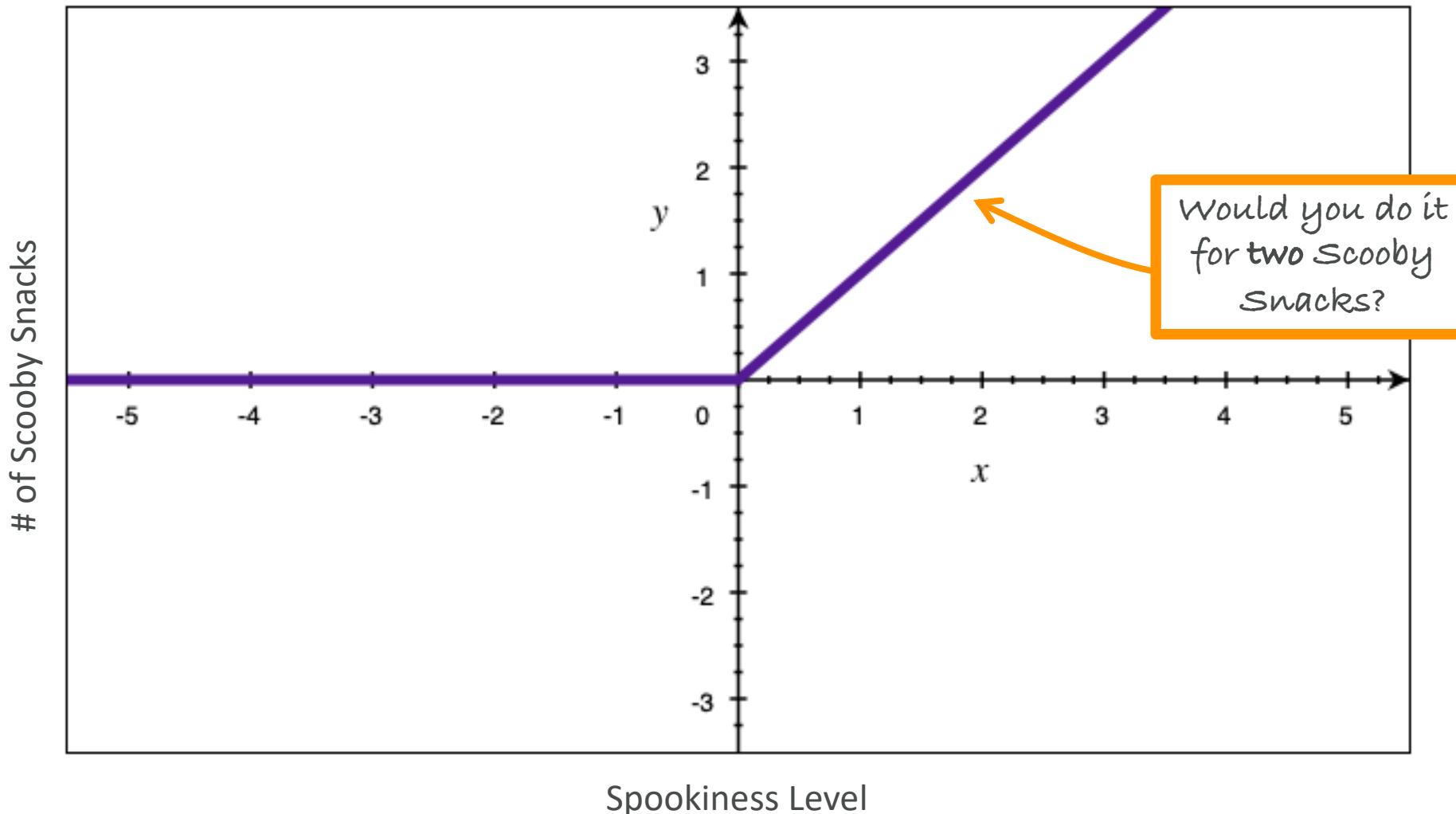
# WHAT ARE NEURAL NETWORKS?



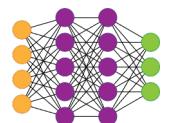
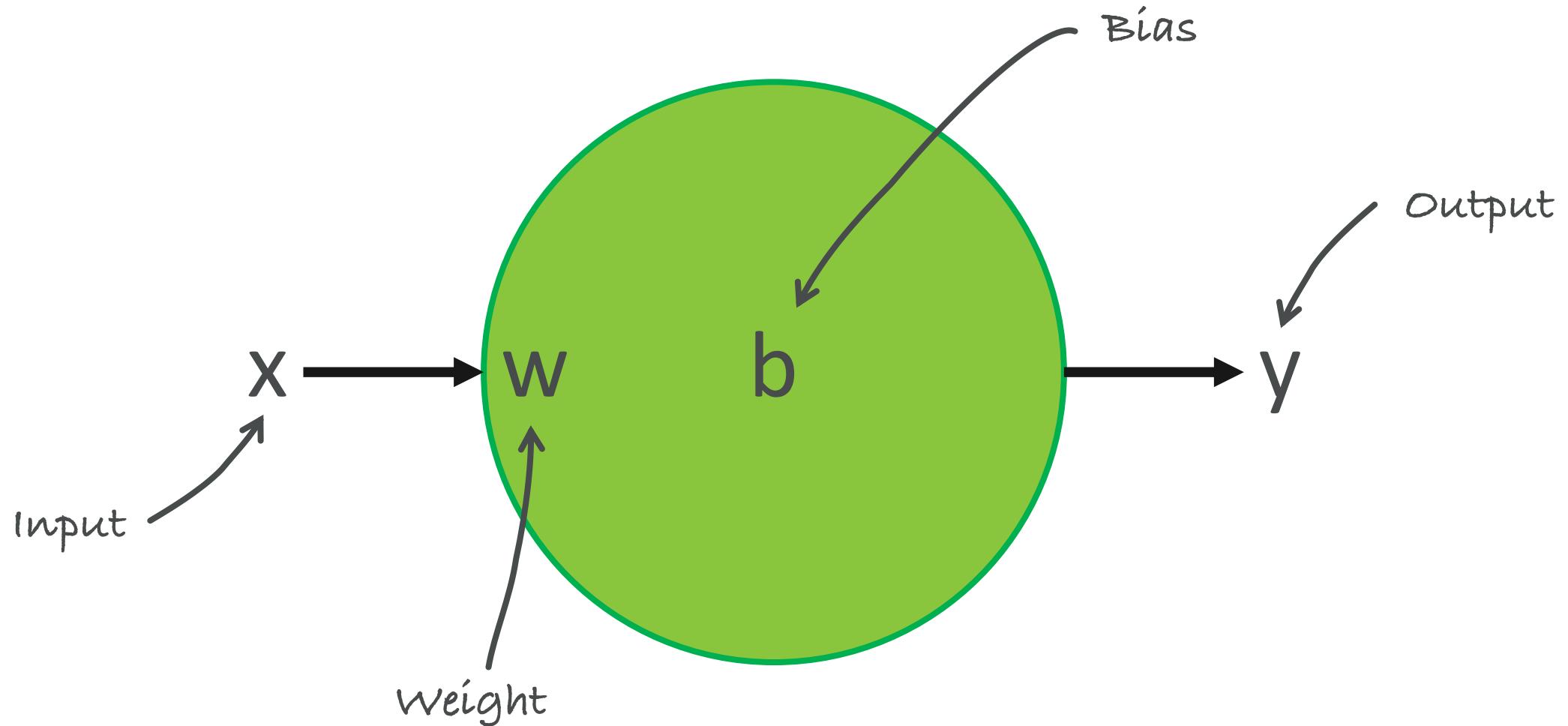
# HOT DOG / NOT HOT DOG



# SCOOBY SNACK REQUIREMENTS



# A SIMPLE NEURON



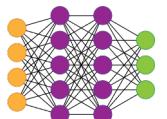
# THE MATH

$$xw + b = y$$

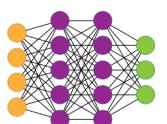
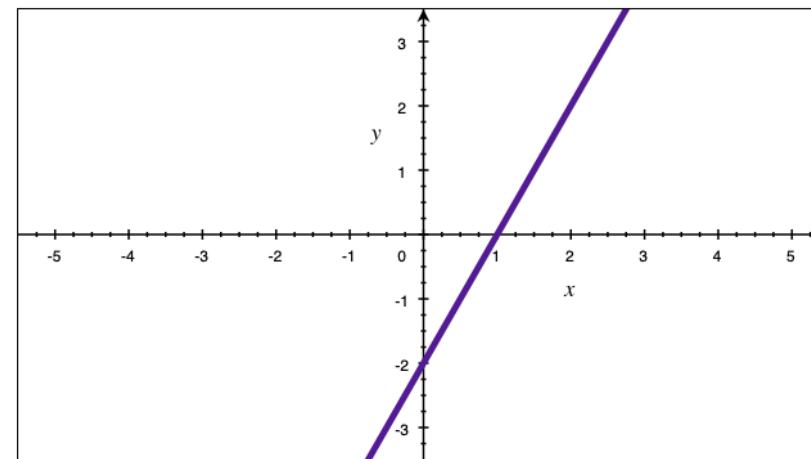
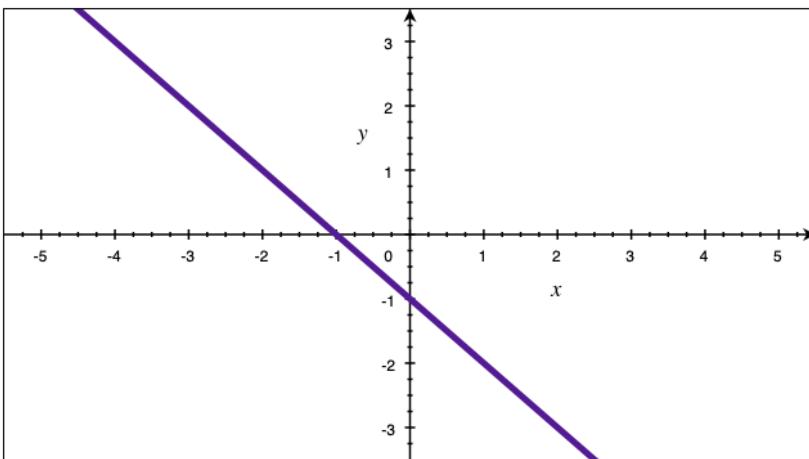
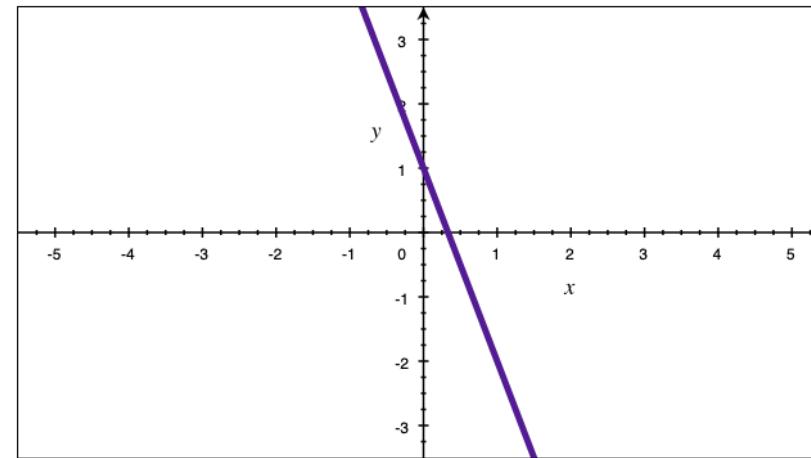
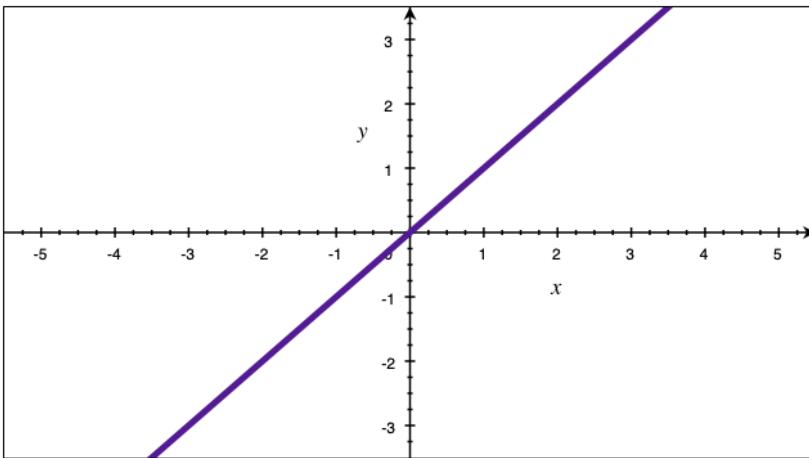
Input →  $xw$  → weight  
Bias →  $b$  → Output

$$y = mx + b$$

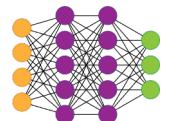
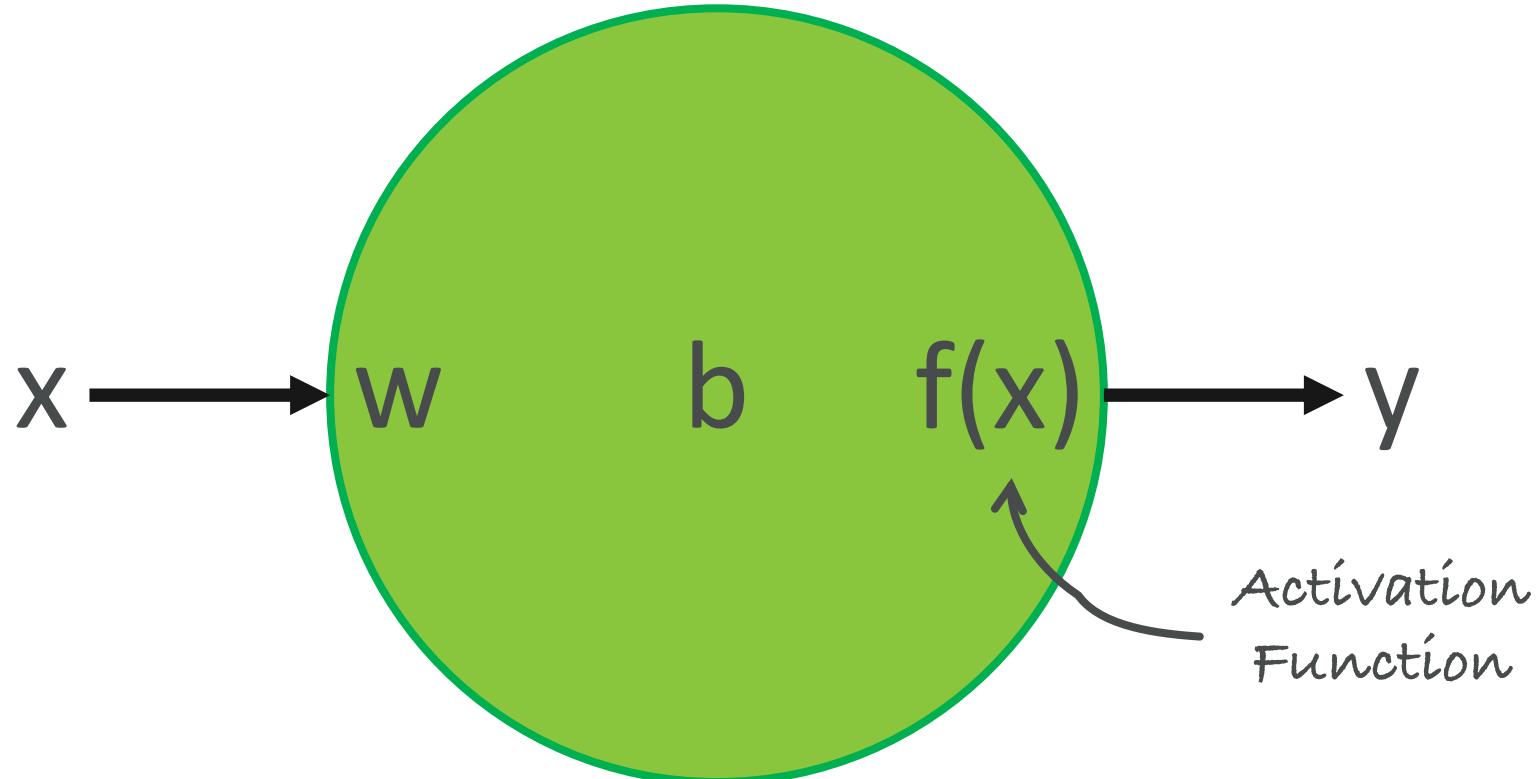
Slope →  $m$   
 $y$ -Intercept →  $b$



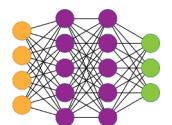
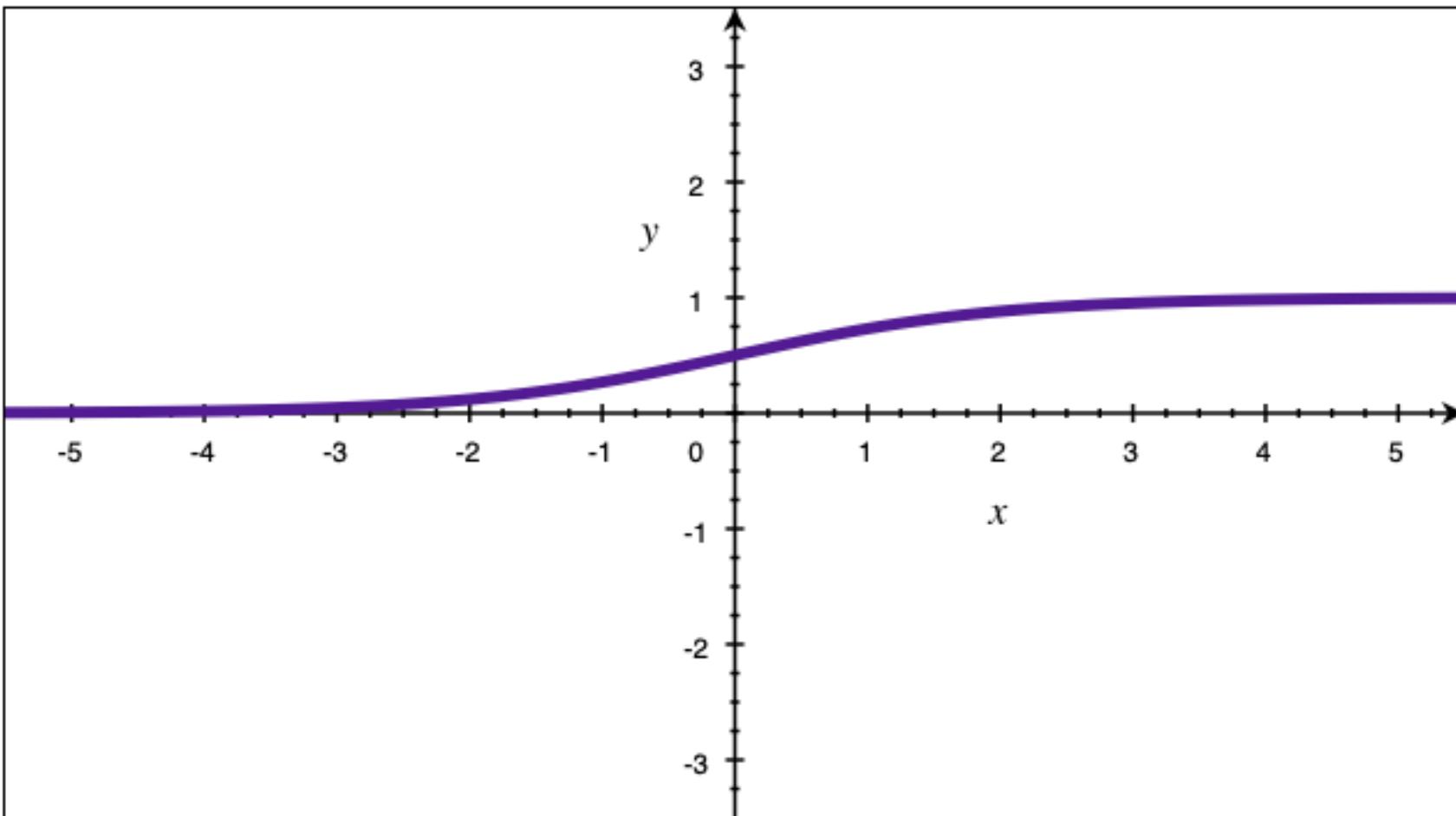
# THE GRAPH



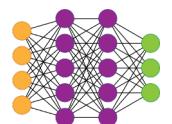
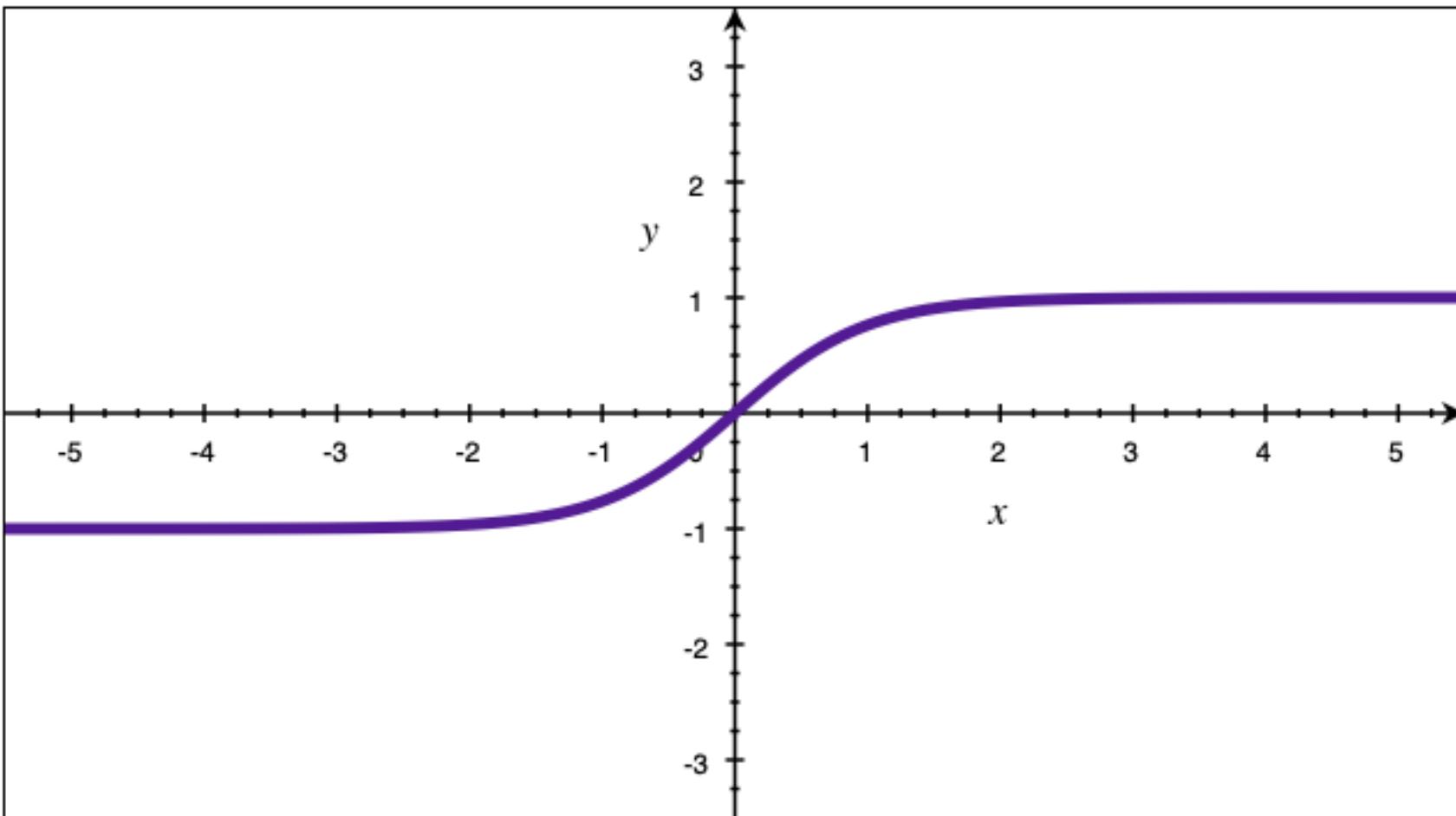
# LESS LINEAR NEURONS



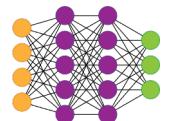
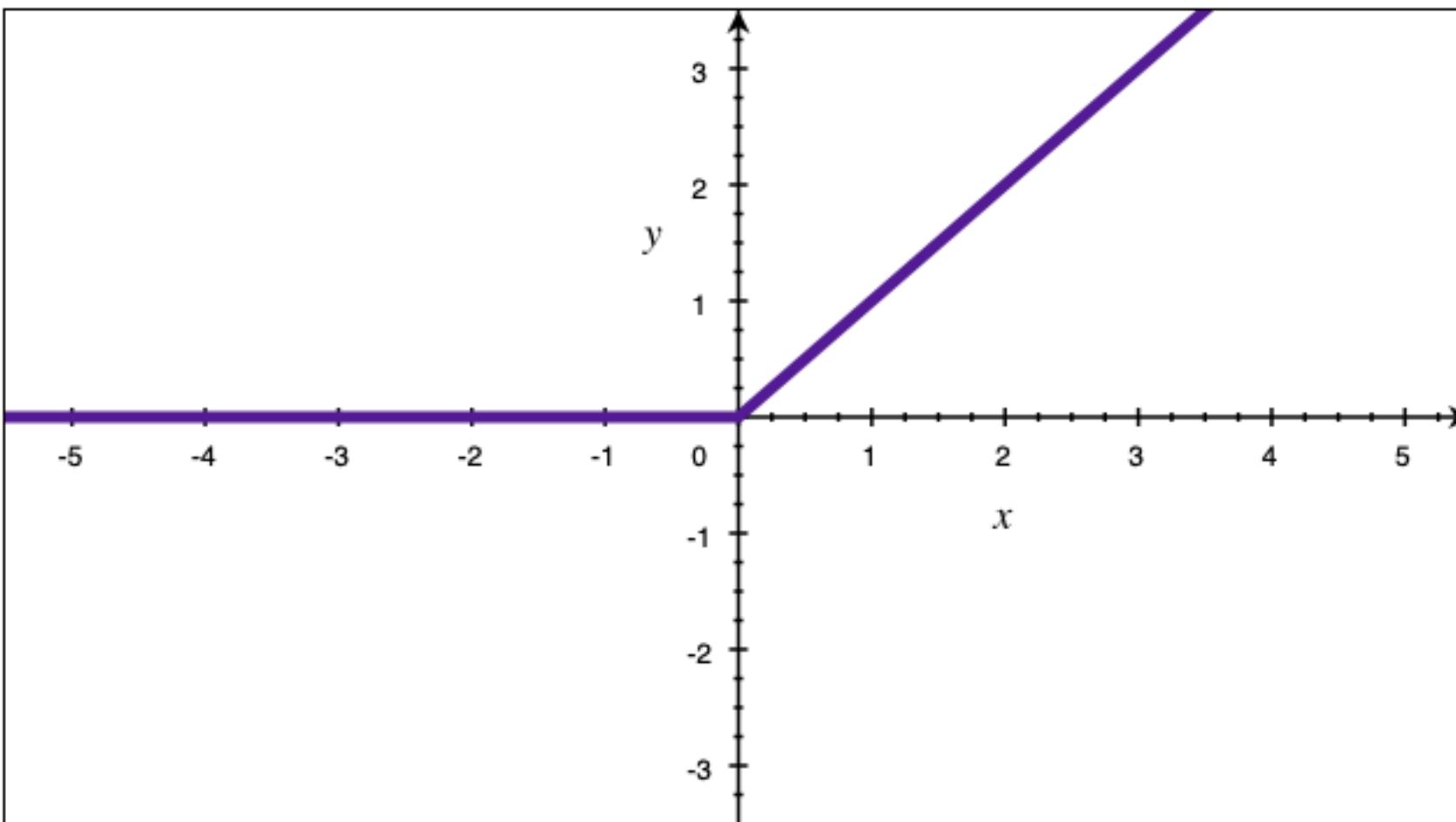
# SIGMOID



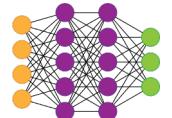
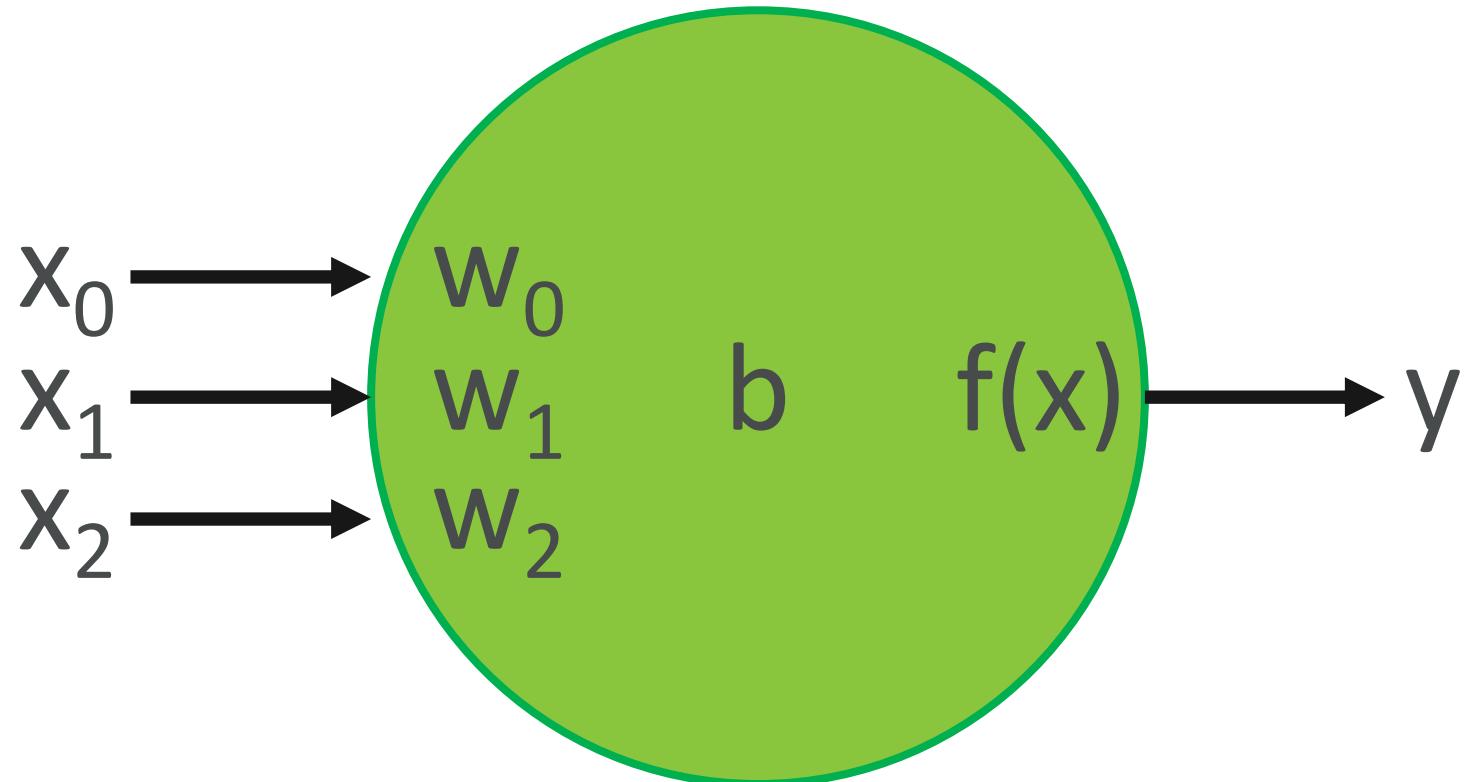
# HYPERBOLIC TANGENT



# RECTIFIED LINEAR UNITS (RELU)



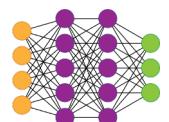
# ANOTHER SCOOBY SNACK?



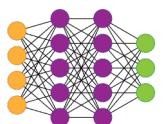
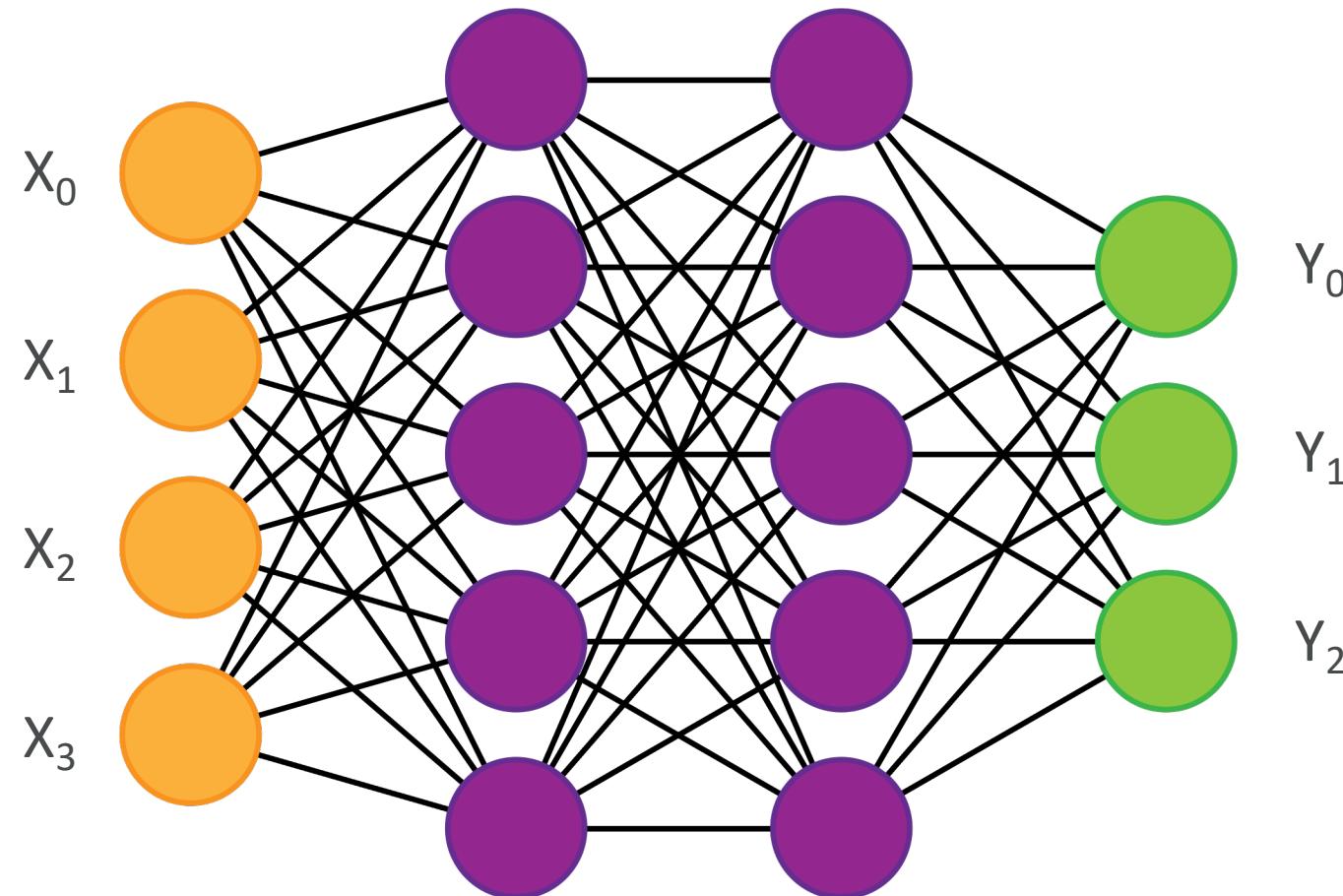
# THE MATH

$$x_0w_0 + x_1w_1 + x_2w_2 + b = y$$

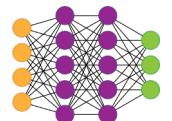
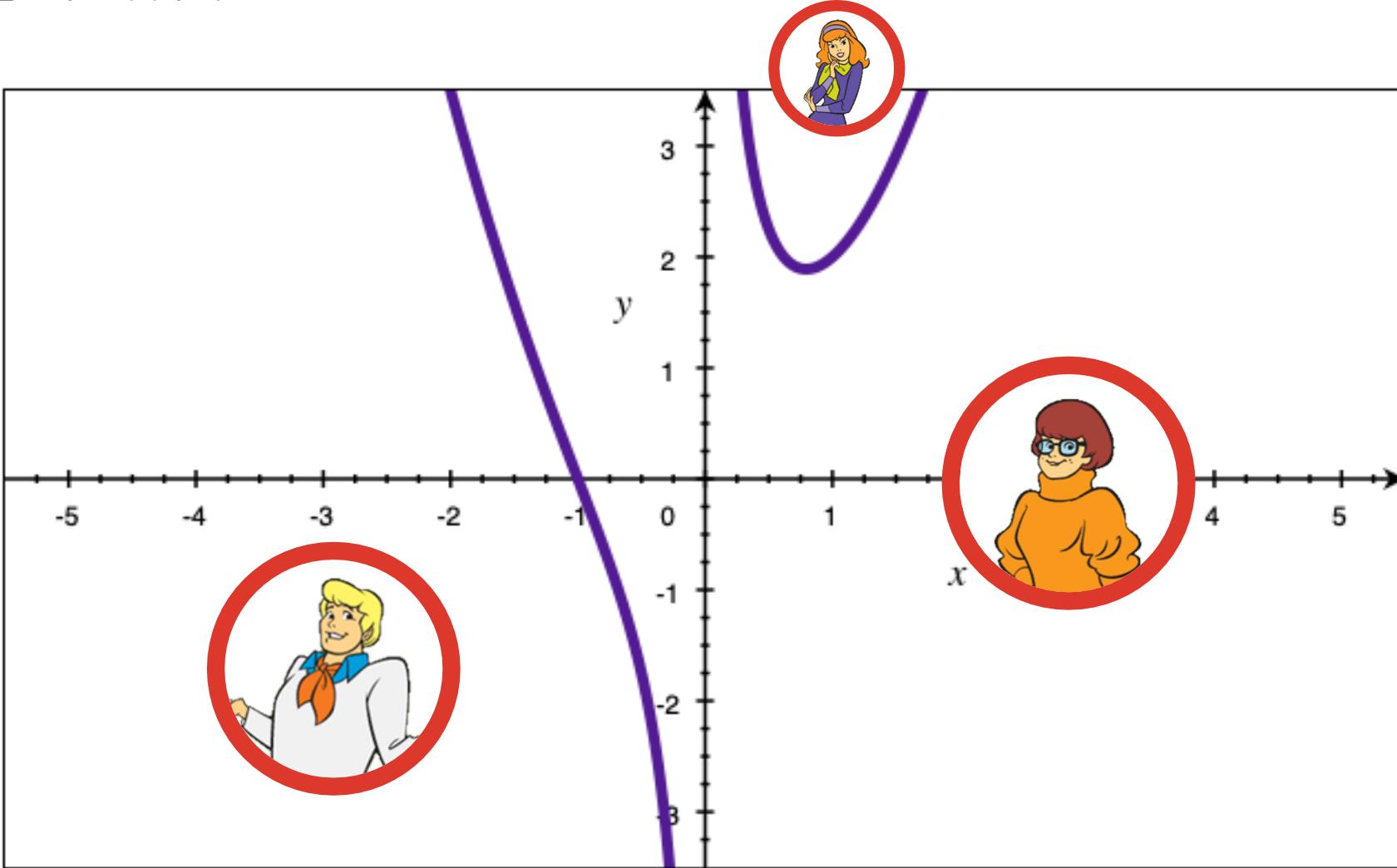
$$\sum_i x_i w_i + b$$



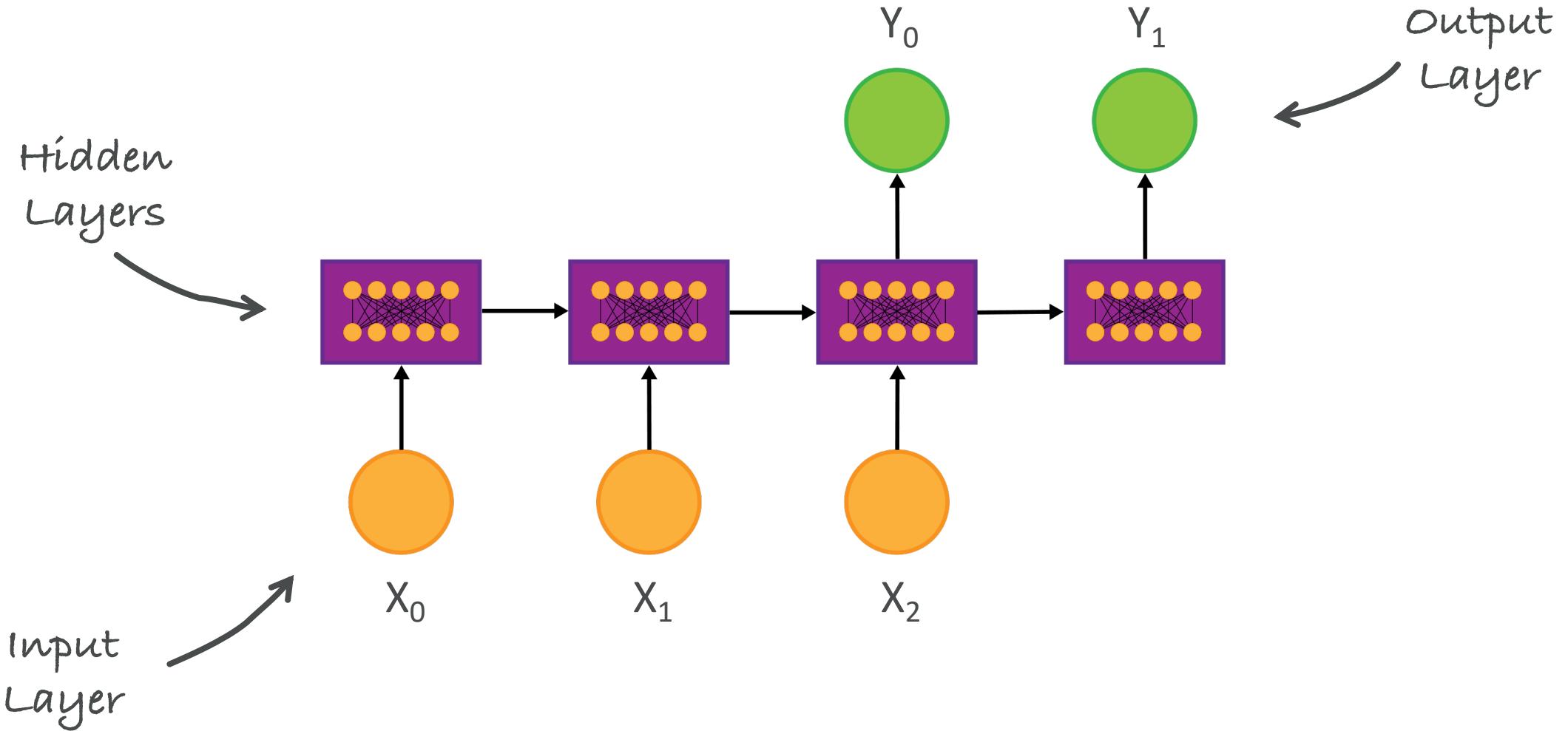
# NEURAL NETWORKS



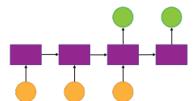
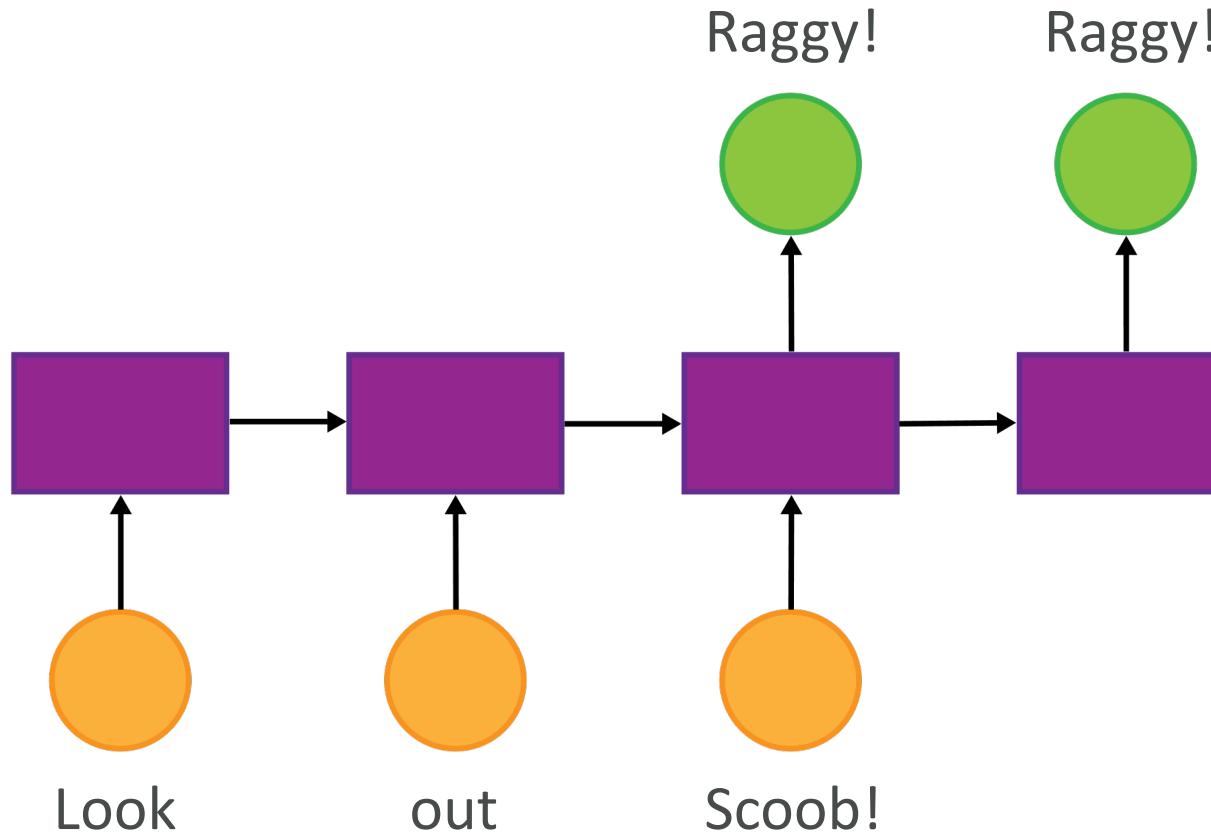
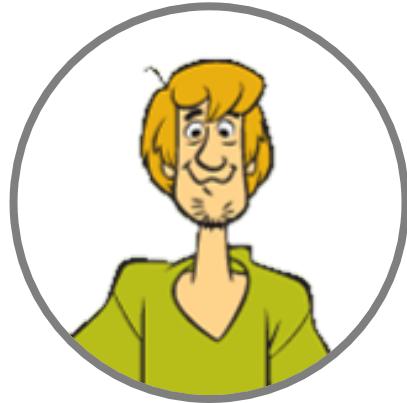
# WHO SAID THAT?



# WHAT ARE RECURRENT NEURAL NETWORKS?



# MANY-TO-MANY



# ONE-TO-MANY

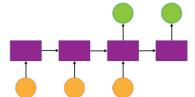
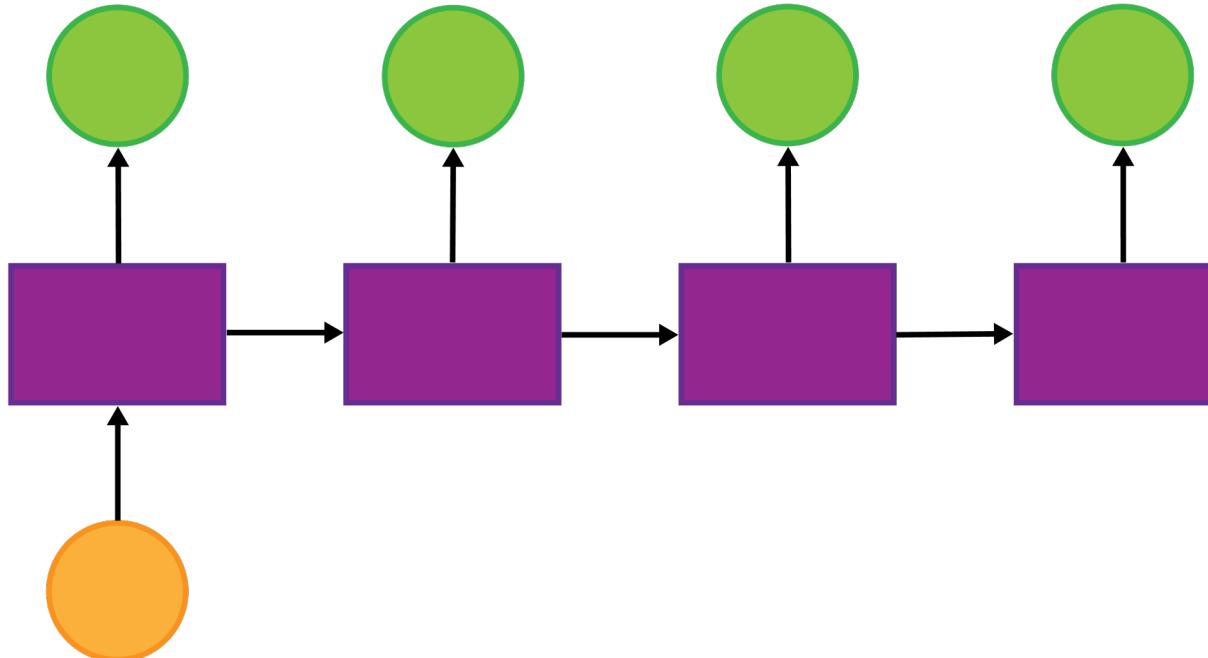


Jeepers!

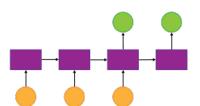
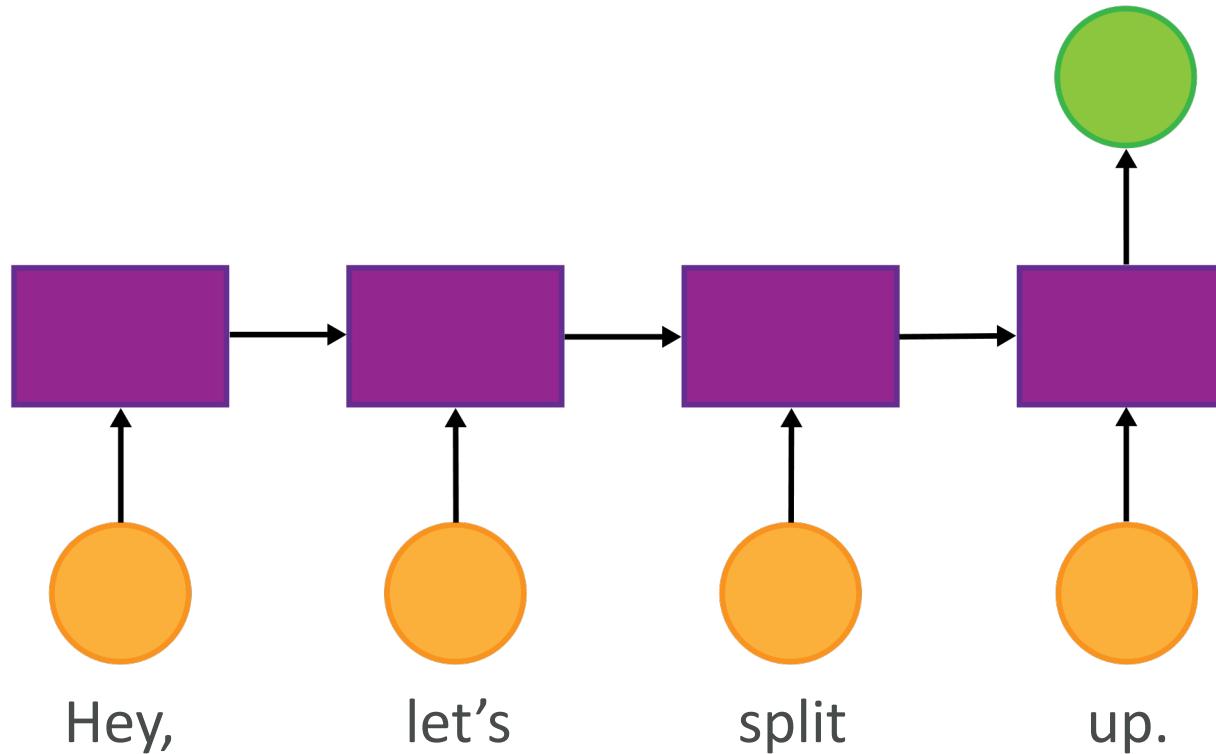
They

mean

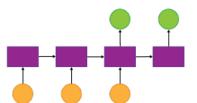
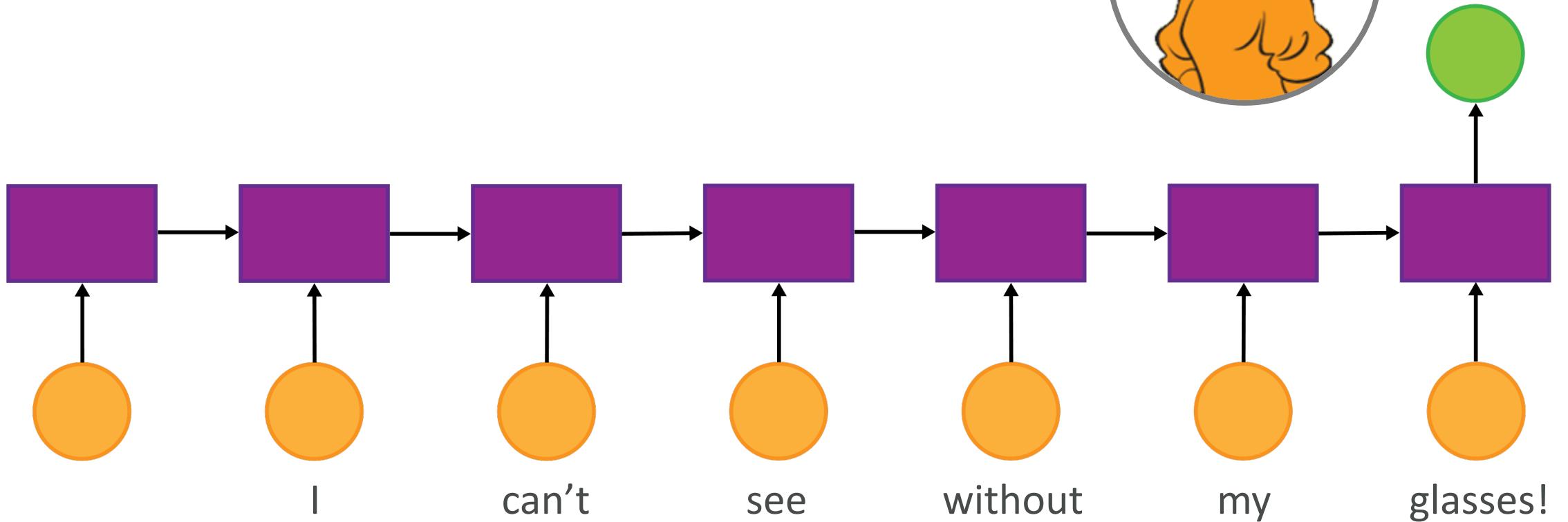
business!



# MANY-TO-ONE



# CLASSIFYING TEXT



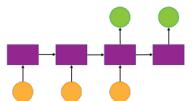
# TURNING WORDS INTO NUMBERS: TOKENIZING

Daphne	Jeepers!
Fred	All right. Let's go!
Scooby	Raggy! Relp!
Scooby	Right Raggy!
Scooby	Rokay Raggy.
Shaggy	Like, run!
Shaggy	Like, step on it Scoob.
Shaggy	Zoinks! Like, look out Scoob!
Velma	Jinkies!
Velma	My glasses! I can't see without my glasses!



Count	Word	Index
3	like	1
3	raggy	2
2	glasses	3
2	my	4
2	right	5
2	scoob	6
1	all	7
1	cant	8
1	go	9
1	i	10
1	it	11
1	jeepers	12

Count	Word	Index
1	jinkies	13
1	lets	14
1	look	15
1	on	16
1	out	17
1	relop	18
1	rokay	19
1	run	20
1	see	21
1	step	22
1	without	23
1	zoinks	24



# TURNING WORDS INTO NUMBERS: TOKENIZING



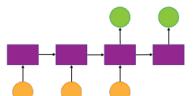
“I can’t see without  
my glasses”



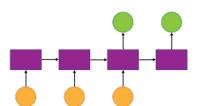
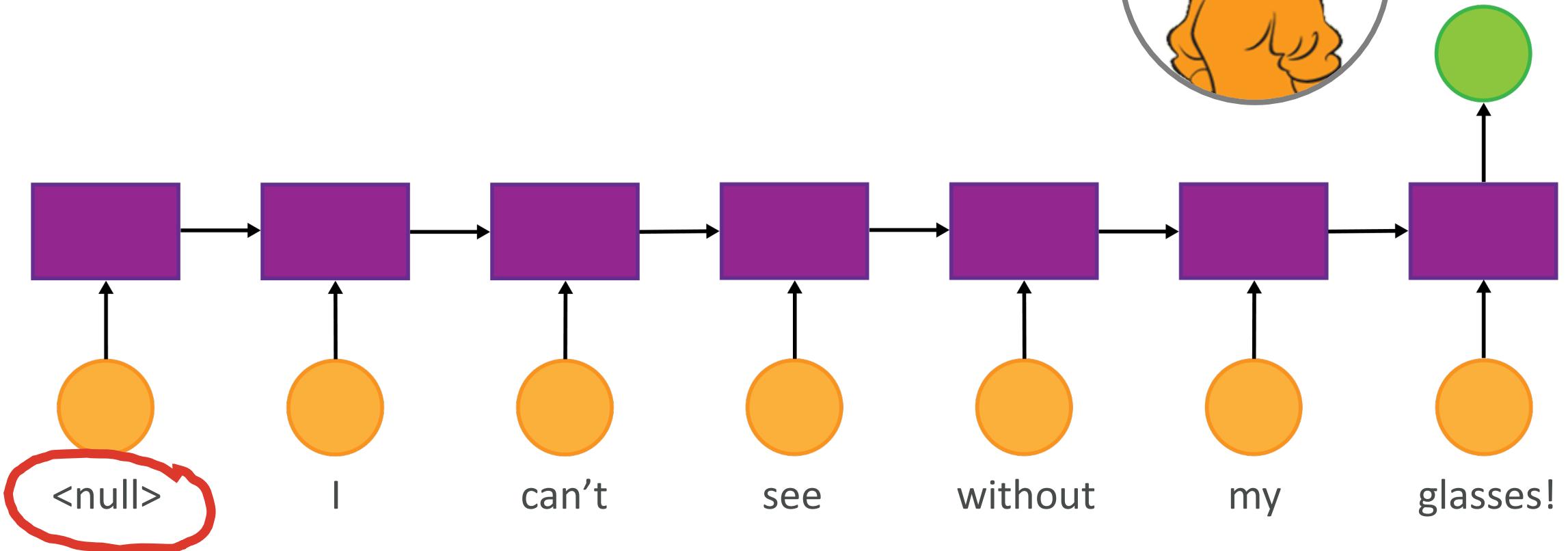
[ 10 8 21 23 4 3 ]

Count	Word	Index
3	like	1
3	raggy	2
2	glasses	3
2	my	4
2	right	5
2	scoob	6
1	all	7
1	cant	8
1	go	9
1	i	10
1	it	11
1	jeepers	12

Count	Word	Index
1	jinkies	13
1	lets	14
1	look	15
1	on	16
1	out	17
1	relp	18
1	rokay	19
1	run	20
1	see	21
1	step	22
1	without	23
1	zoinks	24



# WHAT ABOUT MISSING WORDS?



# TOKENIZING MISSING WORDS



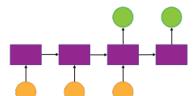
“<null> I can’t see  
without my glasses”



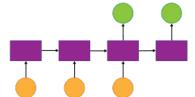
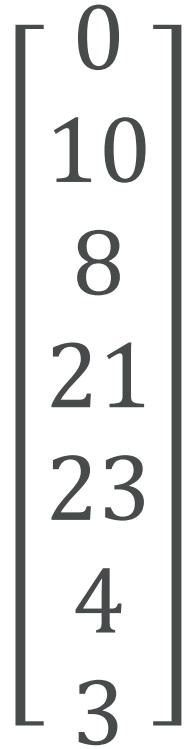
[0 10 8 21 23 4 3]

Count	Word	Index
3	like	1
3	raggy	2
2	glasses	3
2	my	4
2	right	5
2	scoob	6
1	all	7
1	cant	8
1	go	9
1	i	10
1	it	11
1	jeepers	12

Count	Word	Index
1	jinkies	13
1	lets	14
1	look	15
1	on	16
1	out	17
1	relp	18
1	rokay	19
1	run	20
1	see	21
1	step	22
1	without	23
1	zoinks	24



# ONE HOT ENCODING



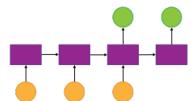
# EMBEDDING



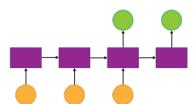
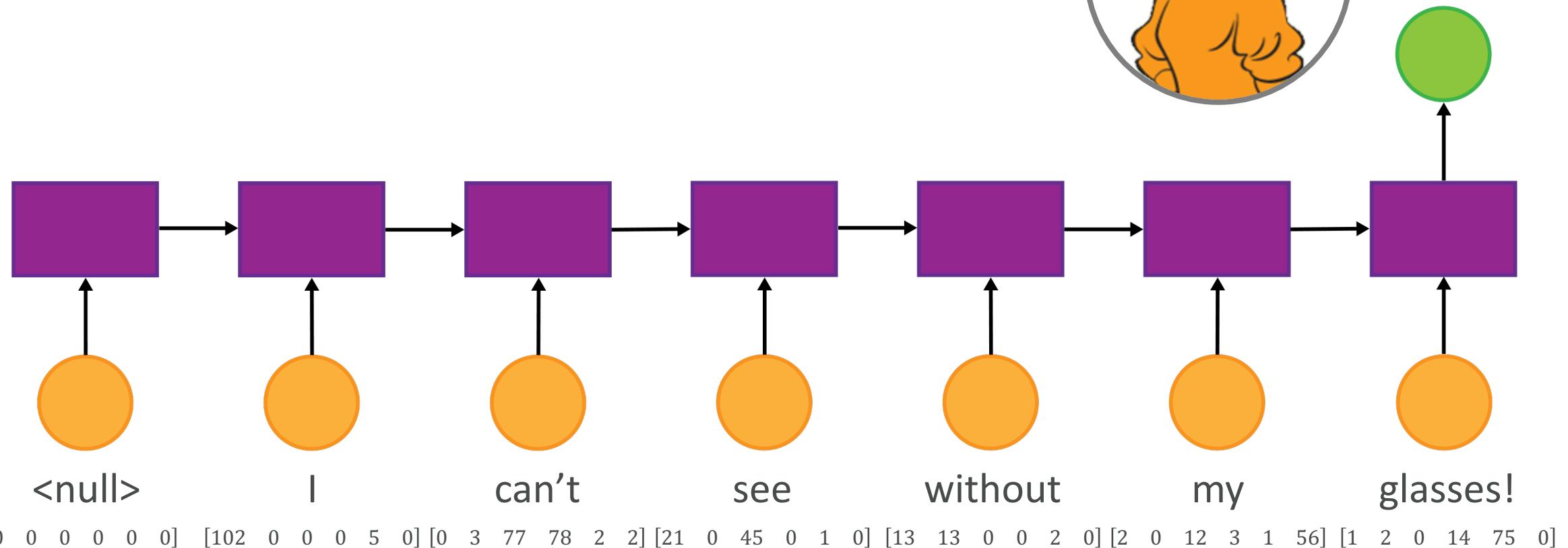
Rows are Words

Columns Encode Semantics

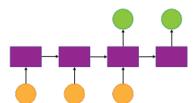
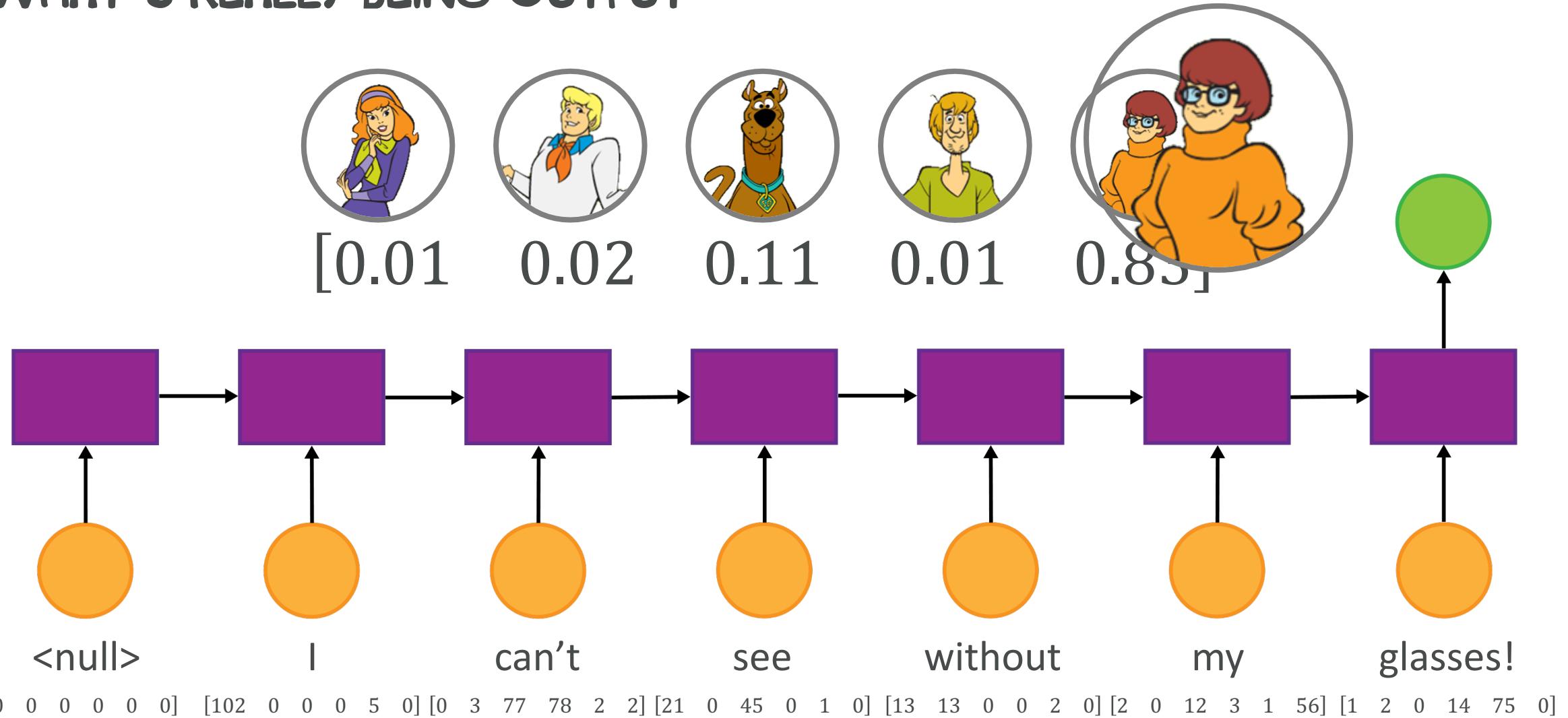
0	like	72	0	7	3	92	0
0	raggy	2	45	11	4	0	1
1	glasses	1	2	0	14	75	0
0	my	2	0	12	3	1	56
0	right	3	2	97	0	0	1
:	...	:	:	:	:	:	:
0	zoinks	13	101	12	9	0	1



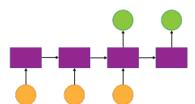
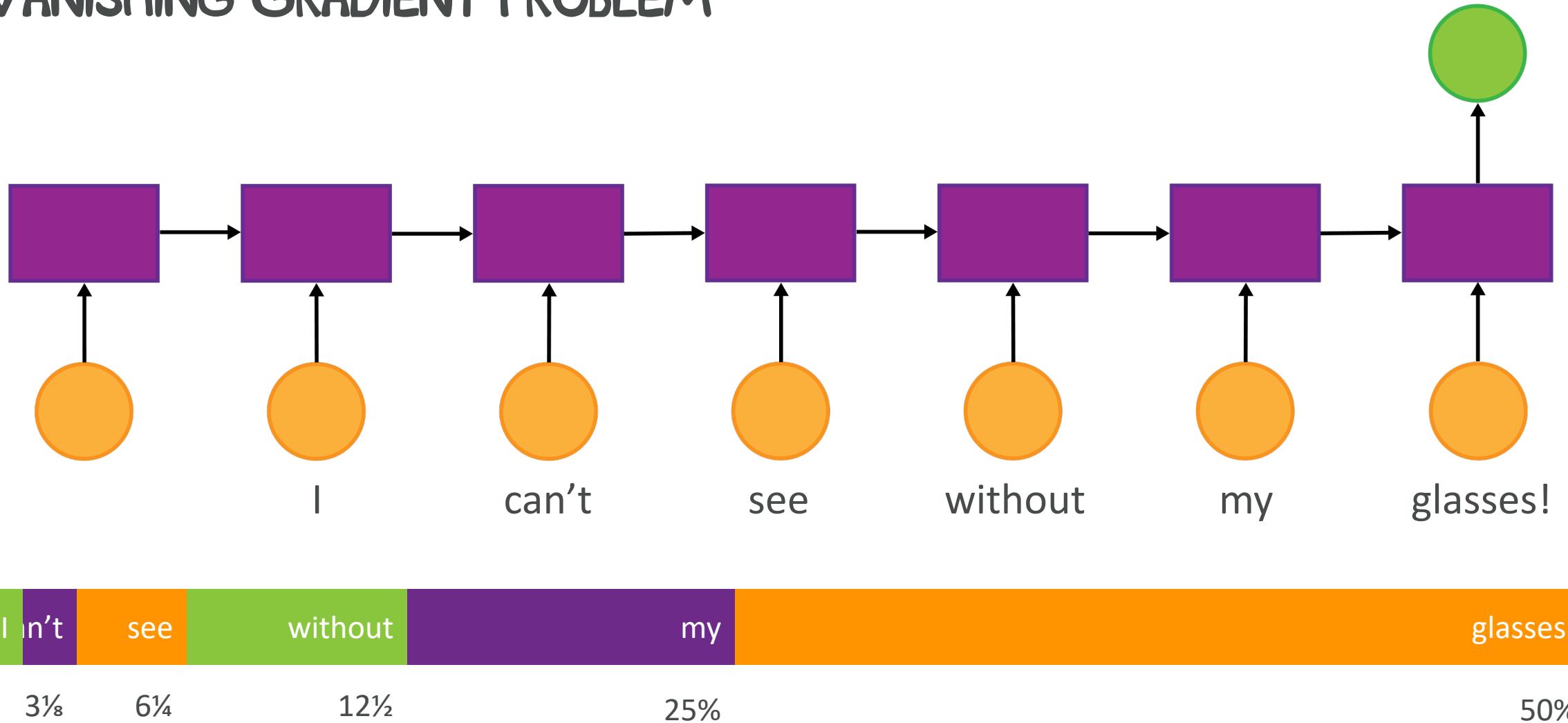
# WHAT'S REALLY BEING INPUT



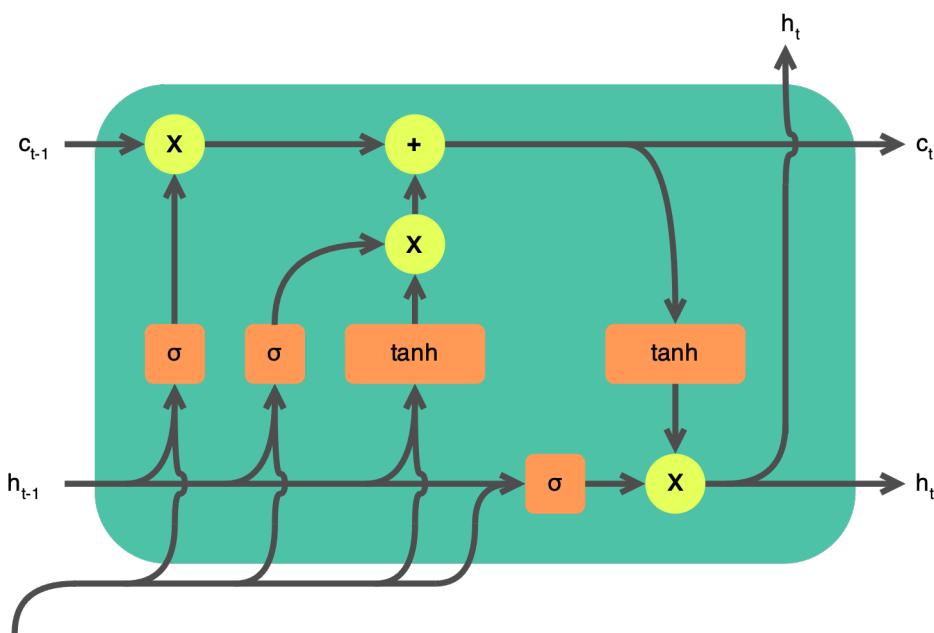
# WHAT'S REALLY BEING OUTPUT



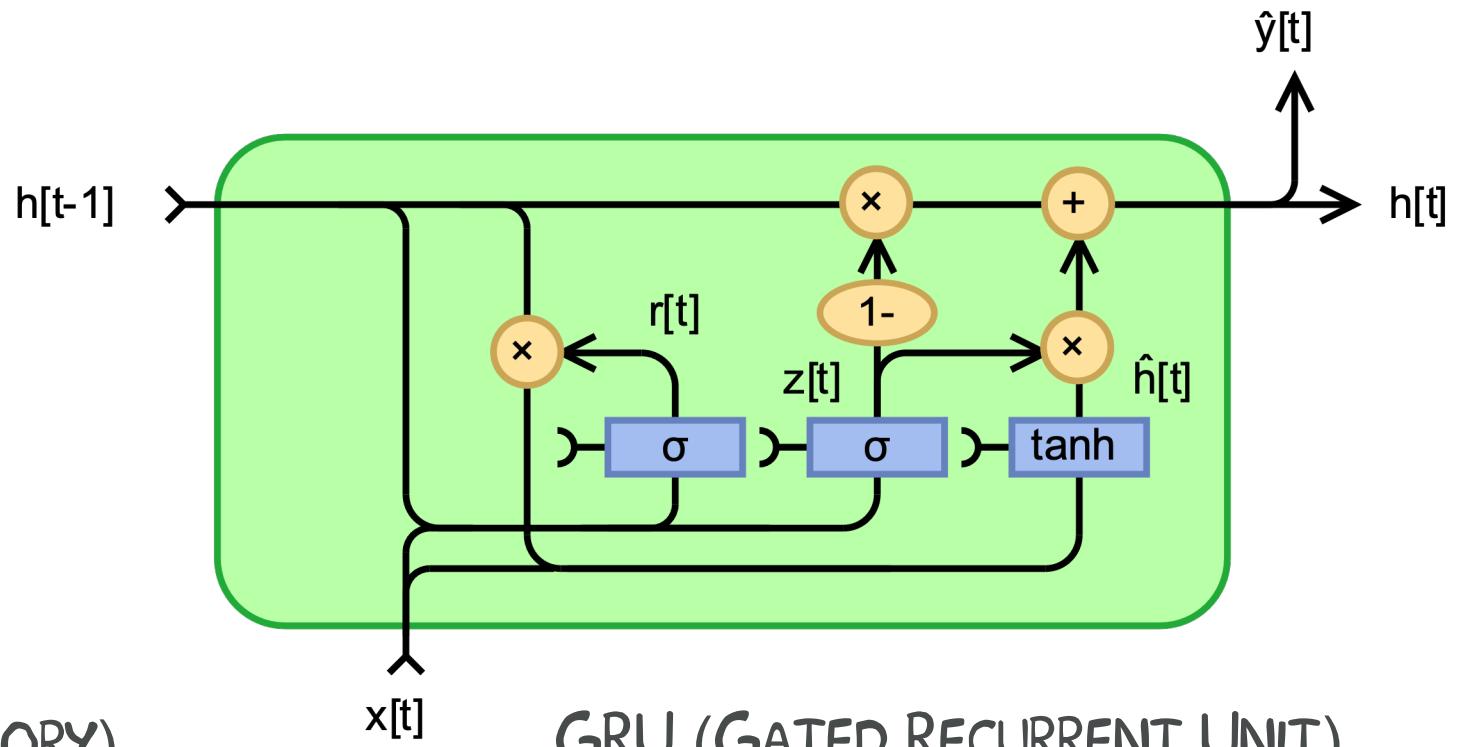
# VANISHING GRADIENT PROBLEM



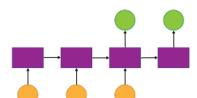
# SOME SOLUTIONS



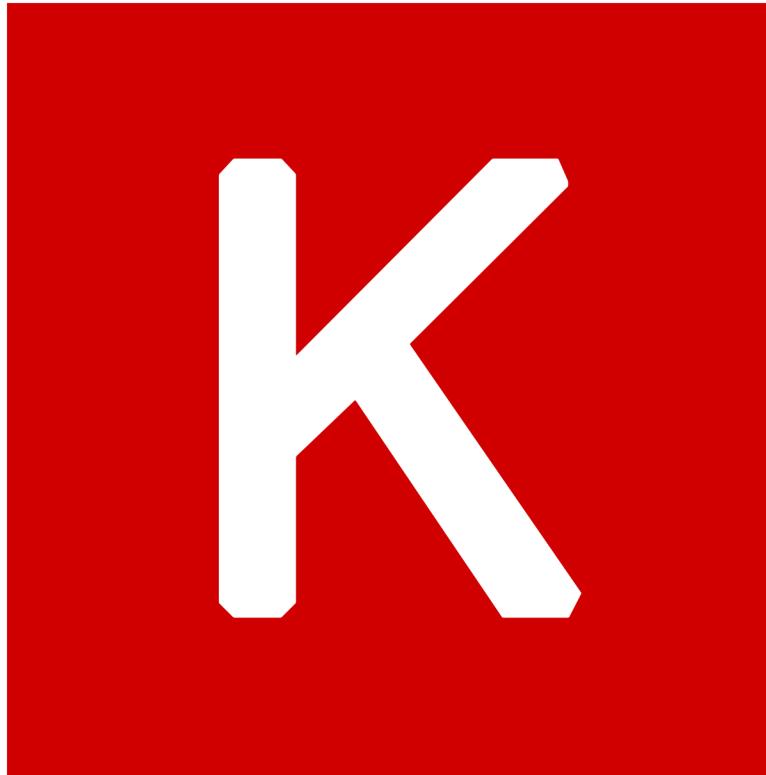
LSTM (LONG SHORT-TERM MEMORY)



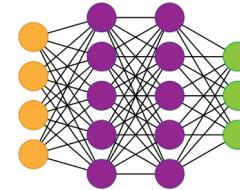
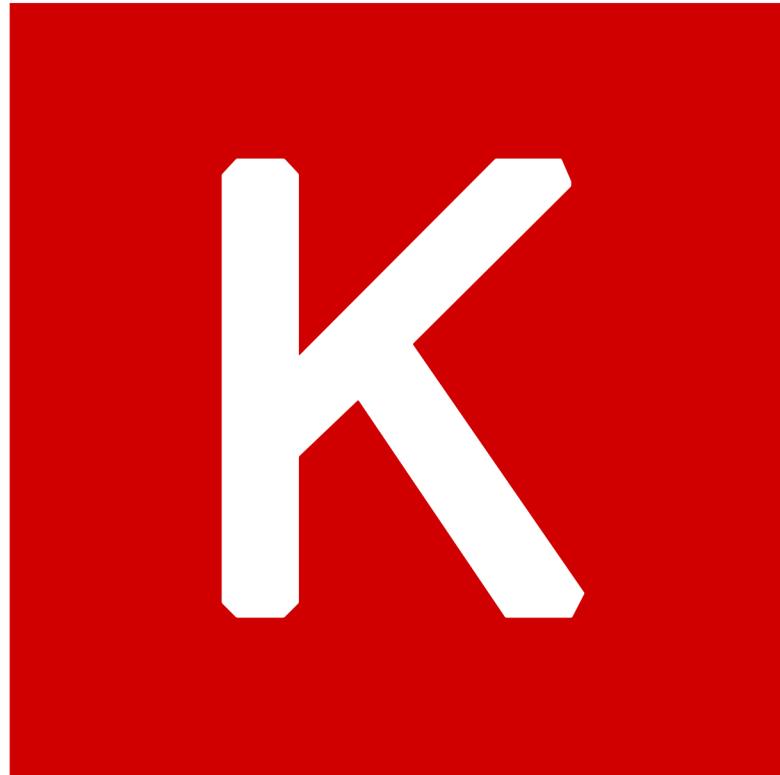
GRU (GATED RECURRENT UNIT)



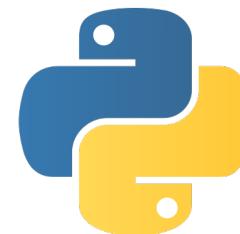
# WHAT IS KERAS?



# WHAT IS KERAS?



DSL that simplifies writing  
neural networks



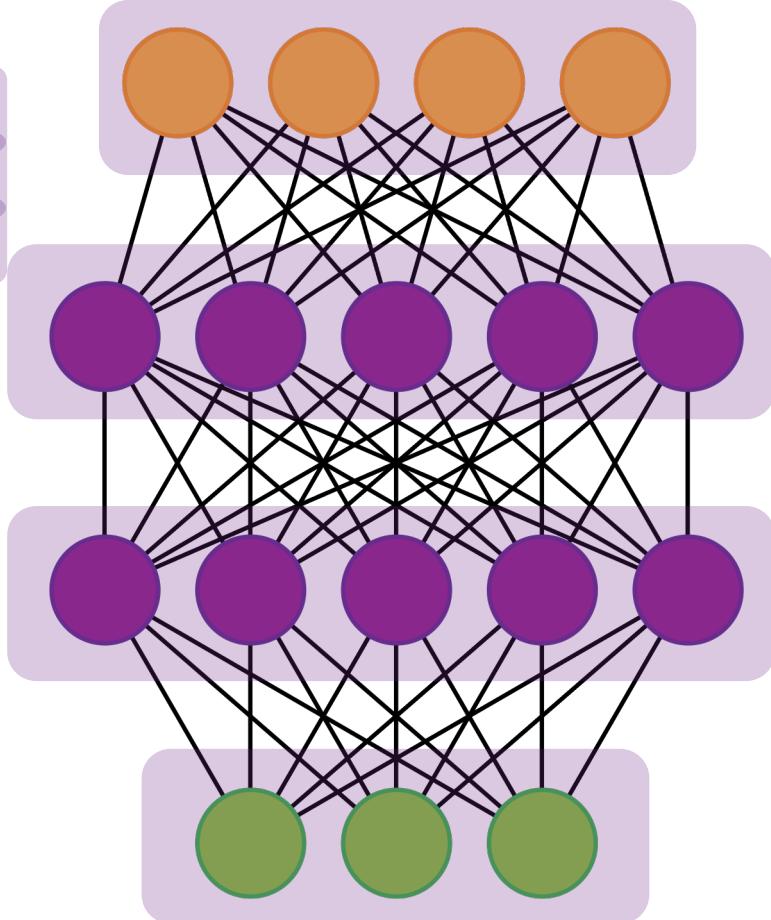
Written for Python  
developers



Recently a part of  
TensorFlow

# DEFINING A SIMPLE NETWORK

```
model = new Sequential()  
  
model.add(Dense(5, activation='relu', input_shape=(4,)))  
model.add(Dense(5, activation='relu'))  
model.add(Dense(3, activation='softmax'))  
  
model.compile(  
    optimizer='adam',  
    loss='categorical_crossentropy',  
    metrics=['accuracy'])  
  
model.fit(X_train, Y_train, batch_size=20, epochs=10)  
  
model.evaluate(X_test, Y_test)  
  
prediction = model.predict(X)
```



# A SIMPLE RNN

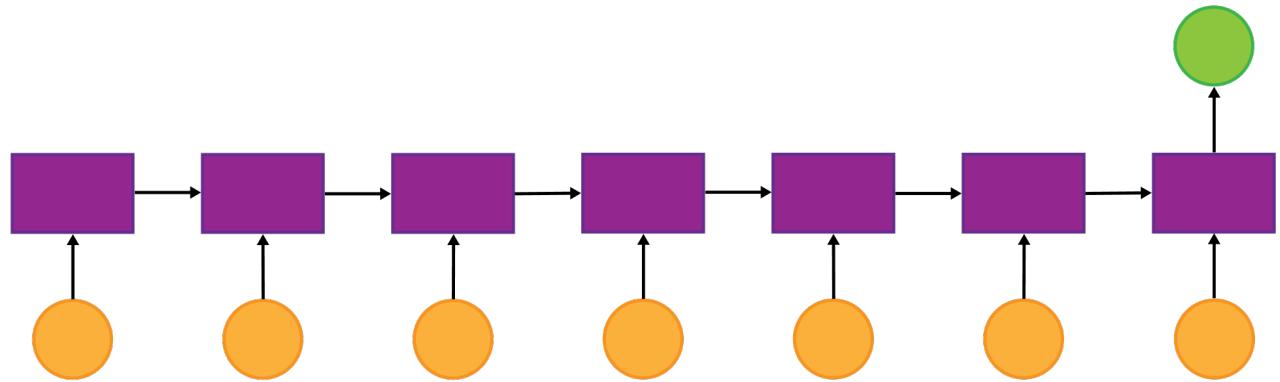
```
MAX_VOCAB = 24  
EMBEDDING_COLUMNS = 6  
MAX_LINE = 7
```

```
model = new Sequential()
```

```
model.add(Embedding(  
    input_dim=MAX_VOCAB,  
    output_dim=EMBEDDING_COLUMNS,  
    input_length=MAX_LINE))
```

```
model.add(SimpleRNN(units=EMBEDDING_COLUMNS))
```

```
model.add(Dense(5, activation='softmax'))
```



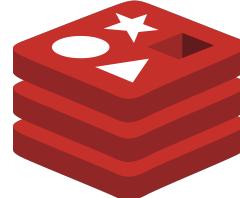


DEM<sup>O</sup>

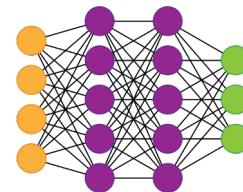
# WHAT IS REDIS AI?



# WHAT IS REDIS AI?



Module that extends  
Redis with AI features



Turns Redis into a  
model server



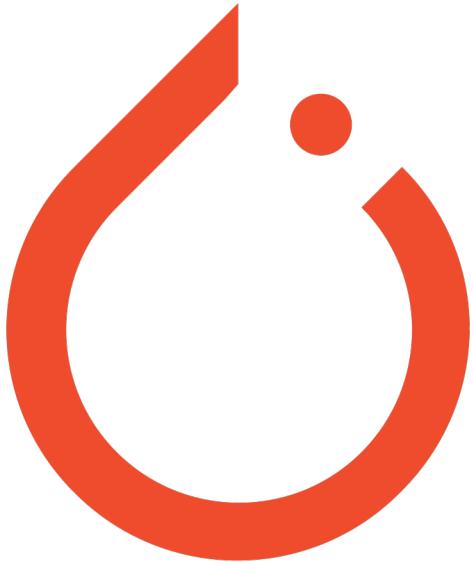
Fast inferences and  
easy integration



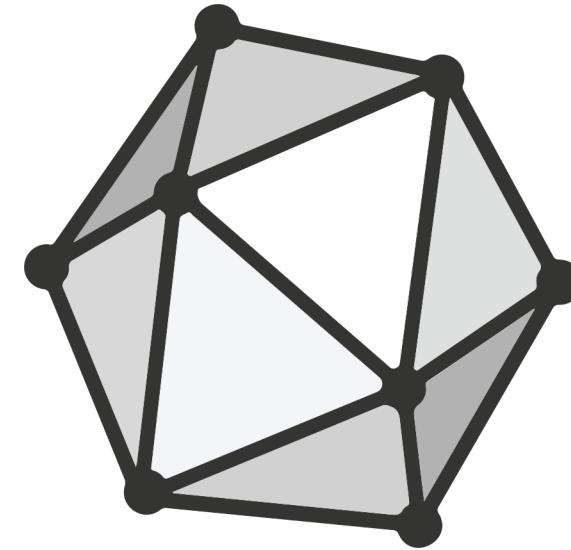
# JINKIES! IT SUPPORTS MULTIPLE BACKENDS



TensorFlow



PyTorch



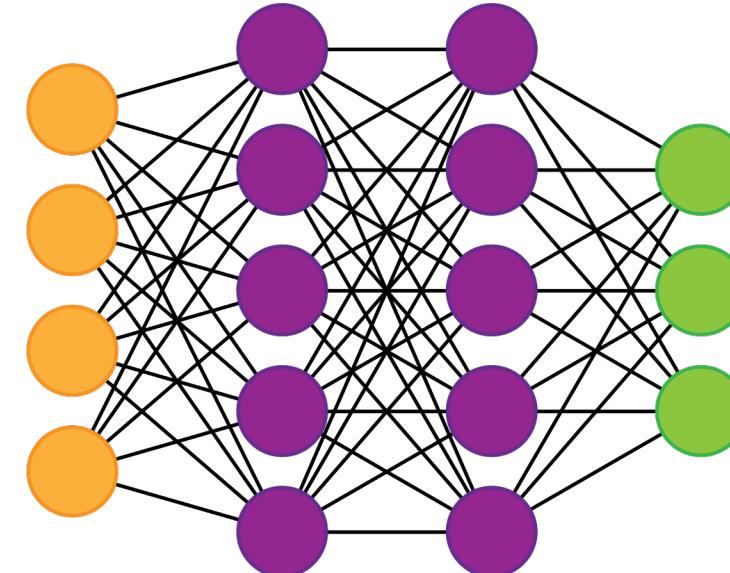
ONNX



# TWO NEW DATA TYPES

$$\begin{bmatrix} 0 & 5 & 6 & 1 & 0 \\ 4 & 3 & 0 & 0 & 2 \\ 7 & 5 & 8 & 2 & 0 \\ 2 & 9 & 1 & 3 & 1 \\ 1 & 7 & 4 & 0 & 5 \end{bmatrix}$$

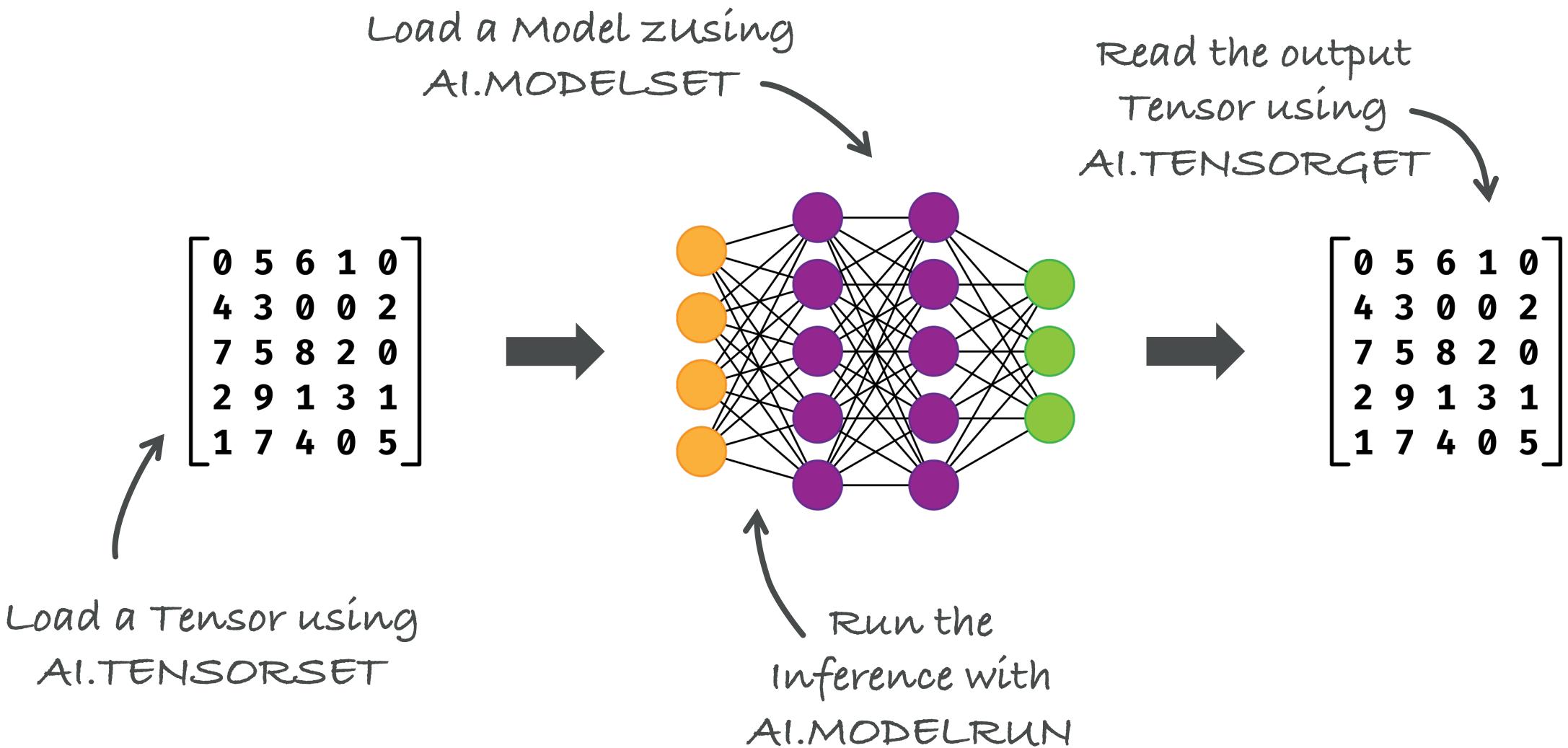
Tensor



Model



# HOW IT WORKS



# WORKING WITH TENSORS

Key →

Type of Number  
FLOAT, DOUBLE, INT8, INT16, INT32,  
INT64, UINT8, UINT16

Shape →

values →

```
> AI.TENSORSET mystery:in FLOAT 2 7
VALUES 0 10 8 21 23 4 3 0 0 24 1 15 17 6
```



"I can't see  
without my  
glasses"



"Zoinks!  
Like, look  
out Scoob!"

Return  
Metadata →

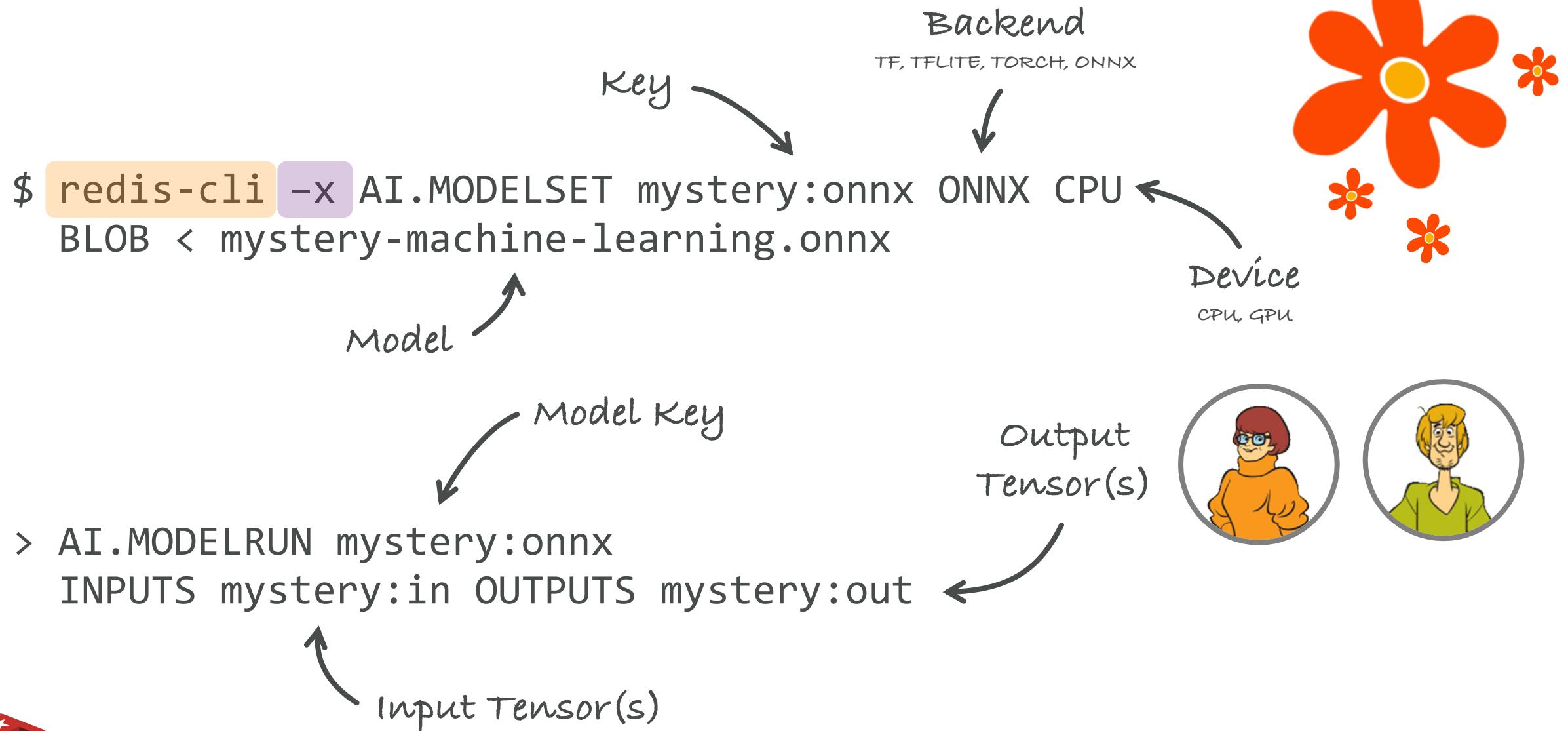
Key →

Return values →

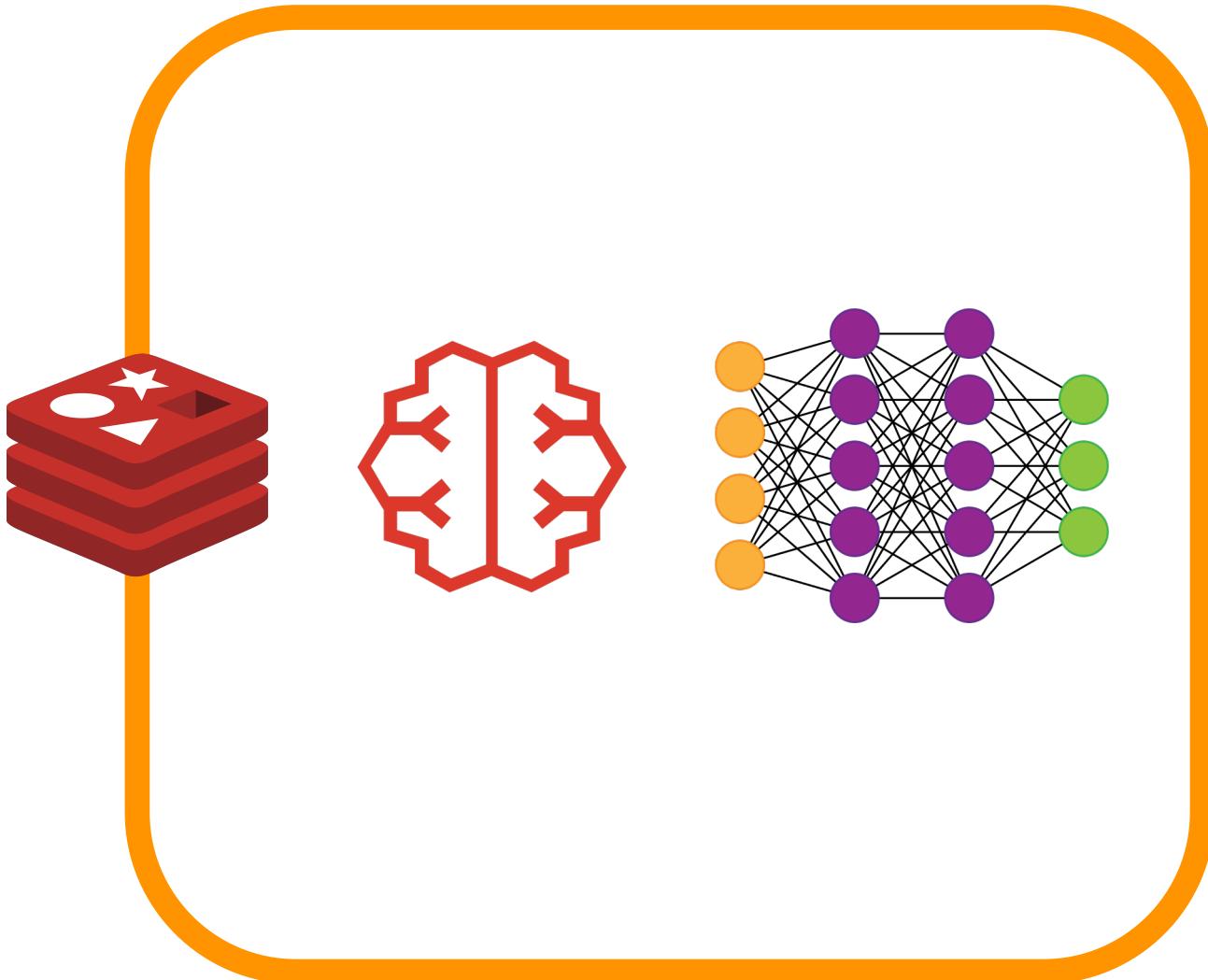
```
> AI.TENSORGET mystery:out VALUES META
```



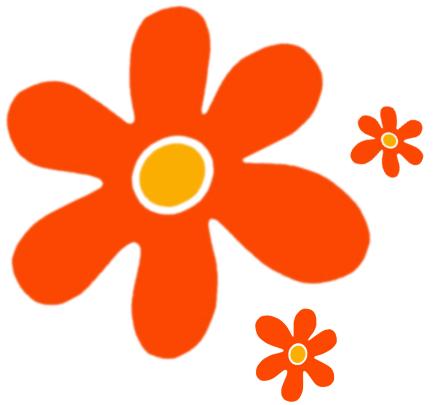
# WORKING WITH MODELS



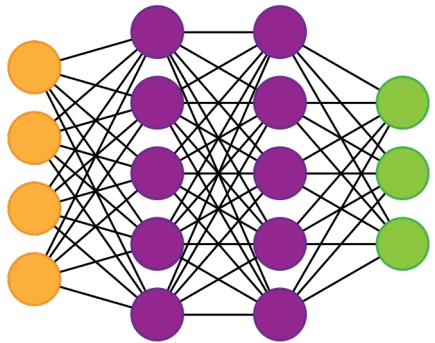
# PUTTING IT ALL TOGETHER



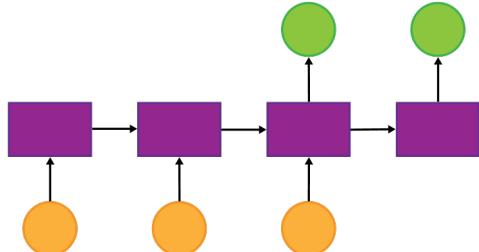
# DEMO



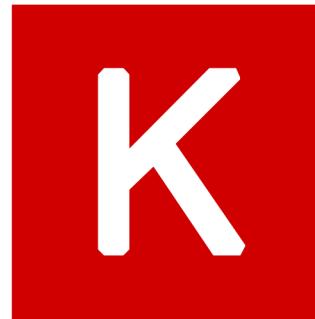
# HAVE YOURSELF A SCOOBY SNACK!



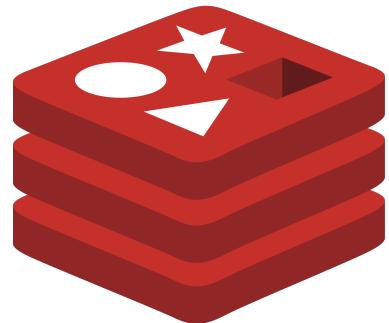
Neural Networks



Recurrent  
Neural Networks



Keras



Redis



## ALL THE LINKS

REDIS

<https://redis.io/>

REDISAI

<http://redisai.io>

DOCKER IMAGE

<https://hub.docker.com/r/redislabs/redisai>

DOCKER IMAGE

<https://github.com/guyroyse/mystery-machine-learning>

# CHECK US OUT



## REDIS DISCORD SERVER

<https://discord.gg/gmCACHU>



## REDIS UNIVERSITY

<https://university.redislabs.com/>



## REDIS UNIVERSITY VIDEOS

<https://www.youtube.com/c/RedisUniversity>





## CODE & SLIDES

[https://github.com/guyroyse/  
mystery-machine-learning](https://github.com/guyroyse/mystery-machine-learning)



**redis**labs  
HOME OF REDIS

---

# Guy Royse

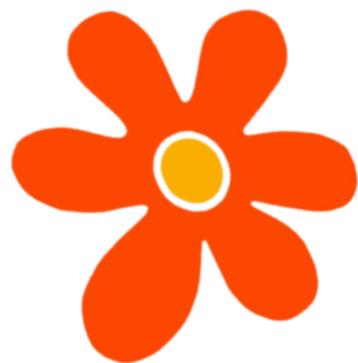
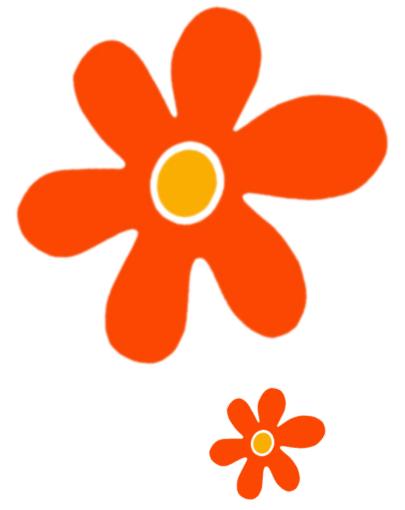
## Developer Advocate

## Redis Labs

 @guyroyse

 [github.com/guyroyse](https://github.com/guyroyse)

 [guy.dev](https://guy.dev)



RHANKS RAGGY!