


TeMS: 17 Lines of Code That Got Me My Dream Job

Dalya Gartzman

TeMS: 17 Lines of Code That Got Me My Dream Job

Dalya Gartzman

**TeMS -
Textual Math Solver**
Simplisico

 **seematics**
VISIONARY
DEEP LEARNING

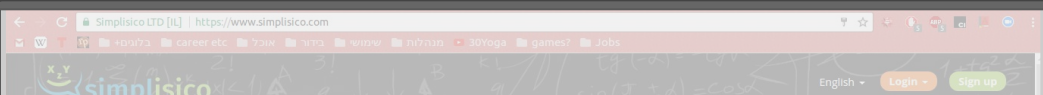




PART I - What is TeMS?

PART I - What is TeMS?

The screenshot displays the Simplisico website, which is a private math tutor. The browser's address bar shows the URL <https://www.simplisico.com>. The website's header includes the Simplisico logo, navigation links for 'English', 'Login', and 'Sign up', and a list of languages: 'עברית', 'אנגלית', 'רוסית', 'ספרדית', 'גרמנית', 'צרפתית', 'איטלקית', 'פורטוגלית', 'הולנדית', 'גרמנית', 'צרפתית', 'איטלקית', 'פורטוגלית', 'הולנדית'. The main content area features a large chalkboard background filled with various mathematical formulas, including trigonometric identities like $\sin(\alpha \pm \beta) = \sin \alpha \cos \beta \pm \cos \alpha \sin \beta$ and $\cos(\alpha \pm \beta) = \cos \alpha \cos \beta \mp \sin \alpha \sin \beta$, as well as geometric formulas like $V = lwh$ and $S.A. = 2lw + 2lh + 2wh$. A man with a beard and a blue shirt stands on the left side of the chalkboard. In the center, there is a video player with a play button icon. Below the video player, a green button with the text 'Try it for free' is visible. The website's footer contains the text 'Simplisico LTD [IL] | <https://www.simplisico.com>' and a list of languages: 'עברית', 'אנגלית', 'רוסית', 'ספרדית', 'גרמנית', 'צרפתית', 'איטלקית', 'פורטוגלית', 'הולנדית', 'גרמנית', 'צרפתית', 'איטלקית', 'פורטוגלית', 'הולנדית'.

PART I - What is TeMS?



English ▾ דליה גרמון ▾


[Subscribe now](#) to get unlimited access to Simplisico's extended explanations, a full month costs much less than a private tutoring hour!

Enter the equation you have to solve using the keyboard or on-screen buttons, when finished press "Show Solution" and learn how to explore your function.

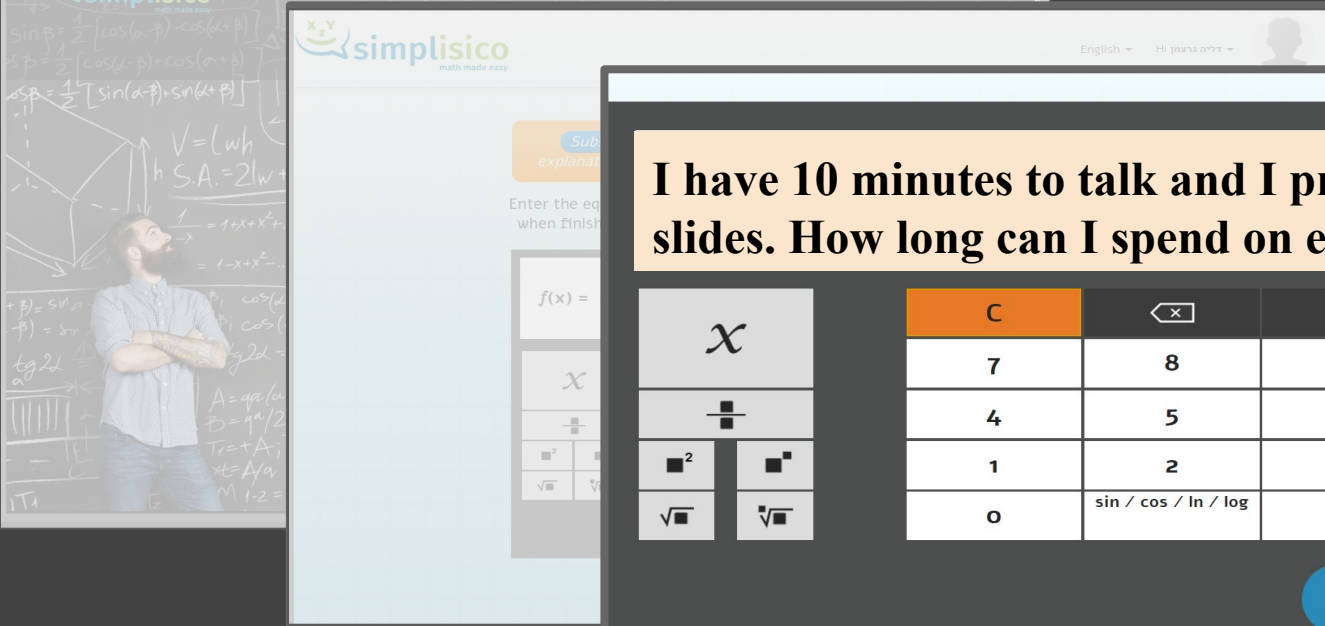
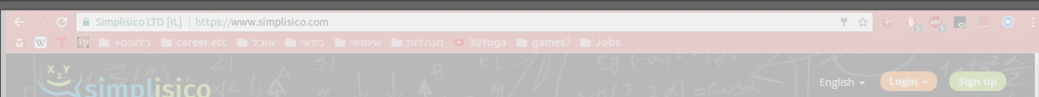
$$f(x) = \frac{5x+6}{2x-3}$$

x	C	\leftarrow	()	\div
$\frac{\Box}{\Box}$	7	8	9	\times
\Box^2 \Box^3	4	5	6	$-$
$\sqrt{\Box}$ $\sqrt[3]{\Box}$	1	2	3	$+$
	0	$\sin / \cos / \ln / \log$	$-$	

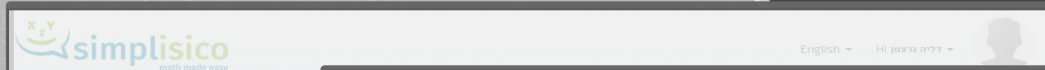
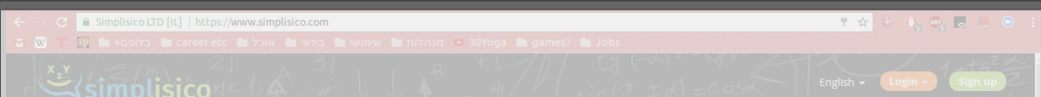
[Show Solution >](#)



PART I - What is TeMS?

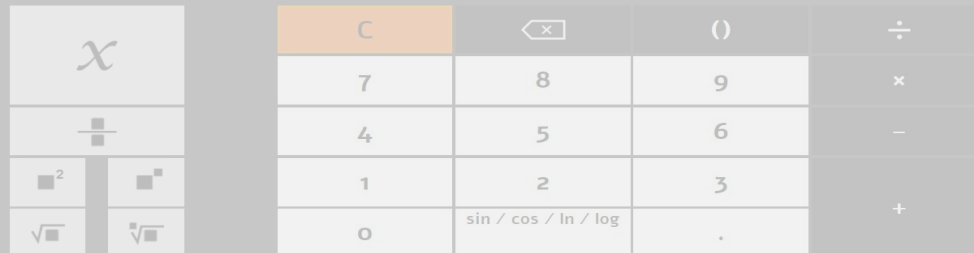


PART I - What is TeMS?



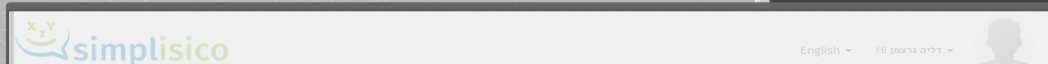
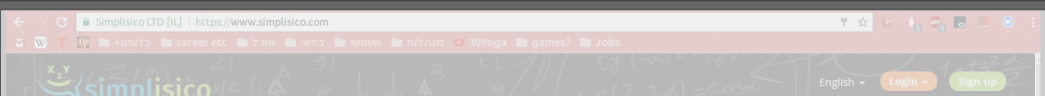
I have 10 minutes to talk and I
slides. How long can I spend on

$$42 * x = 10$$



Show Solution >

PART I - What is TeMS?



Sub
explanat

Enter the eq
when finish

$f(x) =$

x

$\frac{\square}{\square}$

\square^2

$\sqrt{\square}$

I have 10
slides. Ho

x

$\frac{\square}{\square}$

\square^2

$\sqrt{\square}$

\square^2

$\sqrt{\square}$



(What is machine learning?)



(What is machine learning?)

human learning:

shape(●) = ?



(What is machine learning?)

human learning:

If (# \sqcap) = 0
return circle

If (# \sqcap) = 4
return square

shape(\bullet) = ?



(What is machine learning?)

human learning:

If (# \sqcap) = 0
return circle
If (# \sqcap) = 4
return square

shape(\bullet) = ?


machine learning:

shape(\bullet) = ?

(What is machine learning?)

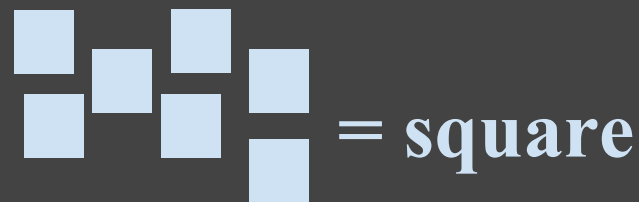
human learning:

If (# ) = 0
return circle

If (# ) = 4
return square

shape() = ?

machine learning:



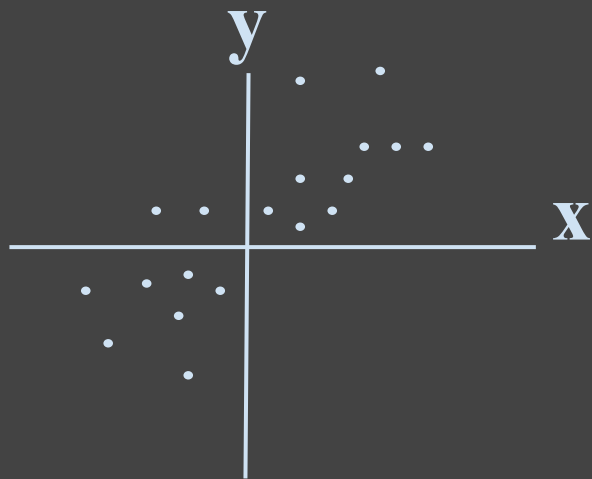
shape() = ?

(What is a neaural network?)



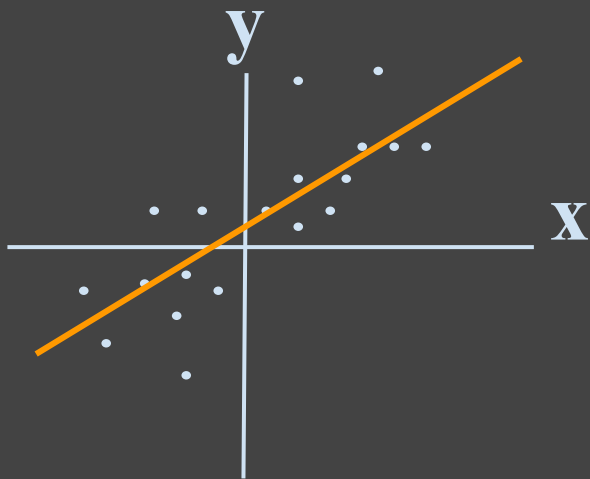
(What is a neural network?)

neuron:



(What is a neural network?)

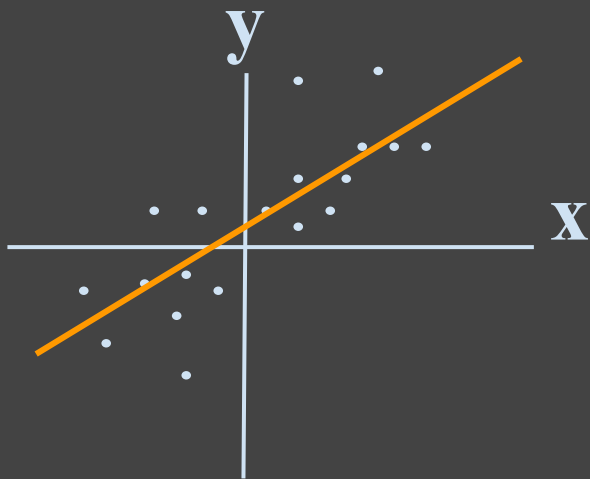
neuron:



$$y=f(x)$$

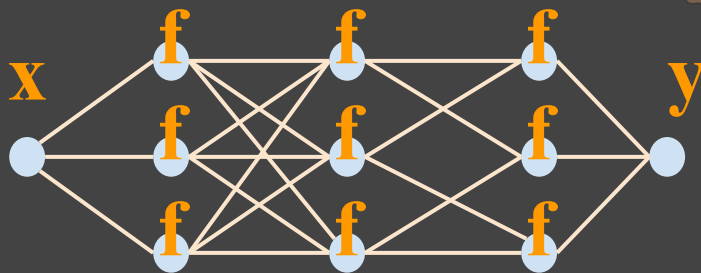
(What is a neural network?)

neuron:



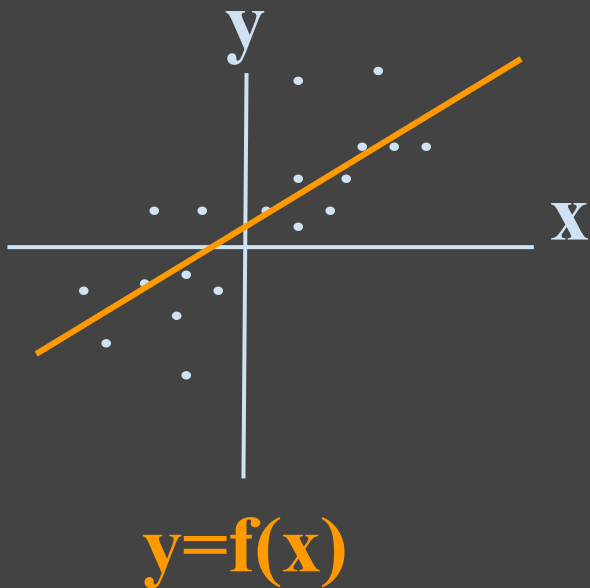
$$y=f(x)$$

neural network:

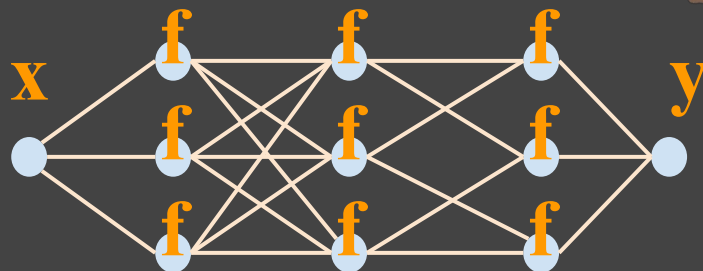


(What is a neural network?)

neuron:



neural network:



- architecture
- what are “f”?
- more stuff...

PART I - What is TeMS?

I have 10 minutes to talk and I prepared 42 slides. How long can I spend on each slide?

$$42 * x = 10$$

\times	C	$\leftarrow \times$	()	\div
$\frac{\square}{\square}$	7	8	9	\times
\square^2 \square^\square	4	5	6	-
$\sqrt{\square}$ $\sqrt[\square]{\square}$	1	2	3	+
	0	sin / cos / ln / log		.

Show Solution >

Preprocessing

Preprocessing

I have 10 minutes to talk and I prepared 42 slides. How long can I spend on each slide?

Preprocessing

I have **A** minutes to talk and I prepared **B** slides. How long can I spend on each slide?

Preprocessing

I have **A** minutes to talk and I prepared **B** slides. How long can I spend on each slide?

TOKENIZE [10, 42, 0] +
[... 'have', '**varA**', 'minutes', ...
'prepared', '**varB**', 'slides', ...]

Preprocessing

I have **A** minutes to talk and I prepared **B** slides. How long can I spend on each slide?

TOKENIZE [10, 42, 0] +
[... 'have', '**varA**', 'minutes', ...
'prepared', '**varB**', 'slides', ...]

EMBED

candy → shop
slides → talk

TeMS: a Sequence-to-Permutation RNN with LSTM and Attention

TeMS: a Sequence-to-Permutation RNN with LSTM and Attention

**I have 10 minutes to talk and
I prepared 42 slides. How
long can I spend on each
slide?**

TeMS: a Sequence-to-Permutation RNN with LSTM and Attention

I have 10 minutes to talk and
I prepared 42 slides. How
long can I spend on each
slide?

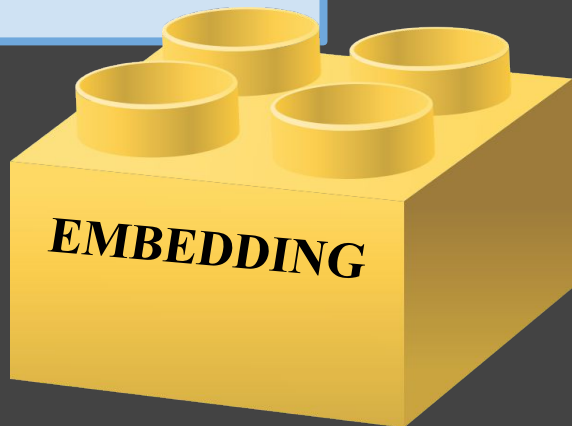
[42, 10, 0]



$42 * x = 10$

TeMS: a Sequence-to-Permutation RNN with LSTM and Attention

I have 10 minutes to talk and
I prepared 42 slides. How
long can I spend on each
slide?



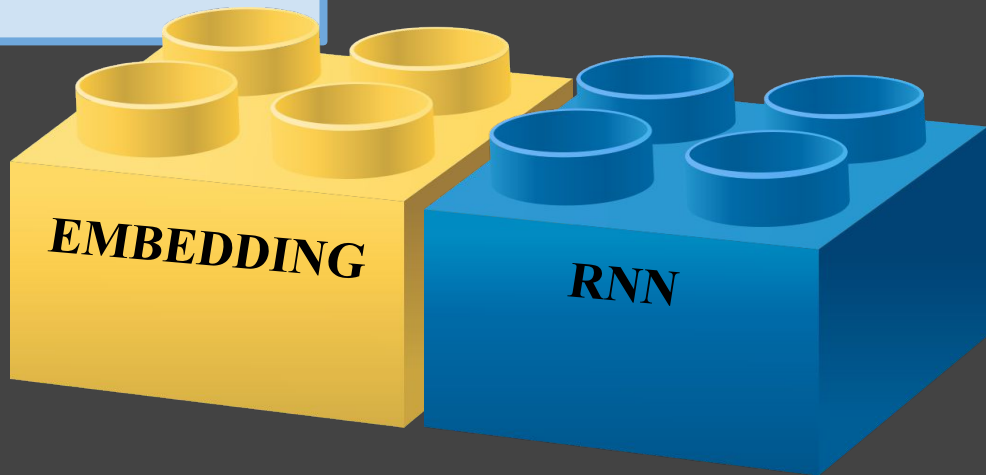
[42, 10, 0]



$42 * x = 10$

TeMS: a Sequence-to-Permutation RNN with LSTM and Attention

I have 10 minutes to talk and
I prepared 42 slides. How
long can I spend on each
slide?



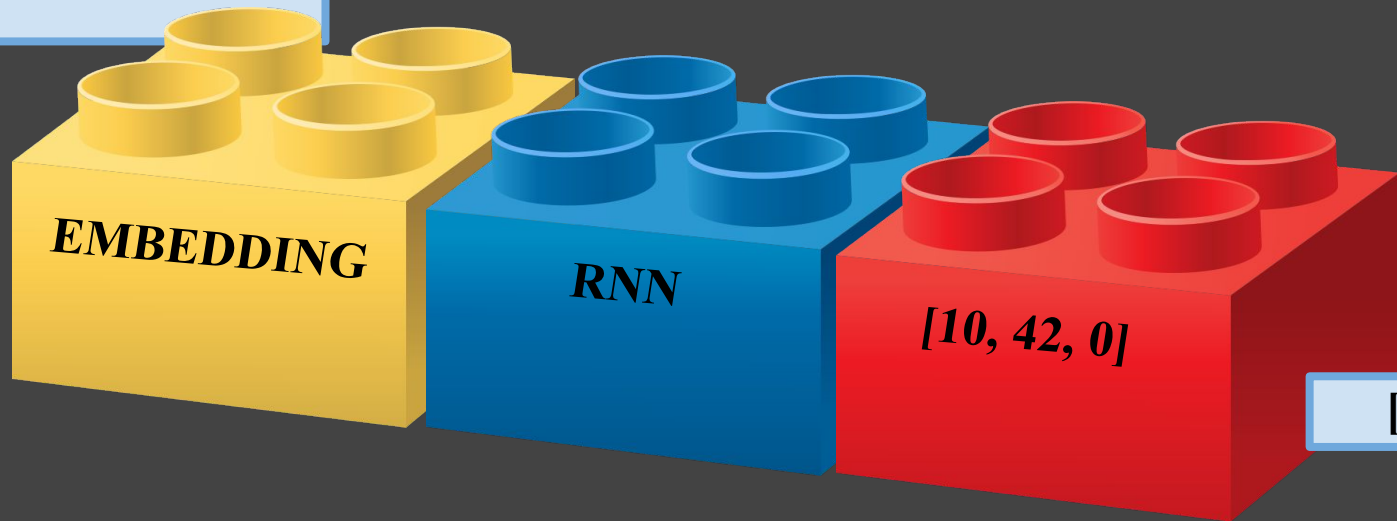
[42, 10, 0]



$42 * x = 10$

TeMS: a Sequence-to-Permutation RNN with LSTM and Attention

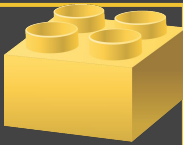
I have 10 minutes to talk and
I prepared 42 slides. How
long can I spend on each
slide?



$[42, 10, 0]$



$42 * x = 10$



```
# generate embeddings
embedding = w2v.Word2Vec()
embedding.build_vocab(all_questions)
embedding.train(all_questions)
```

```
# Words model
words_model = Sequential()
words_model.add(Embedding(vocab_size, embedding_dim, weights=[embedding]))
words_model.add(LSTM(64, activation='softmax', return_sequences=True, dropout=0.5))
words_model.add(LSTM(64, activation='relu', return_sequences=False, dropout=0.5))
words_input = Input(shape=(max_len,))
processed_words = words_model(words_input)
```

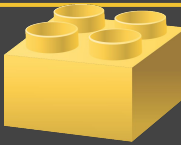
```
# Numbers model
nums_model = Sequential()
nums_model.add(Dense(3, input_dim=3))
nums_input = Input(shape=(3,))
processed_nums = nums_model(nums_input)
```

```
# Concatenate models
merged = keras.layers.concatenate([processed_nums, processed_words])
```

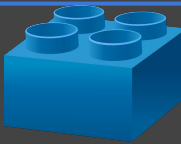
```
# add one dense layer to integrate the merging
hidden = Dense(32, activation='tanh')(merged)
hidden = Dropout(0.5)(hidden)
# finish with a dense layer
output = Dense(3)(hidden)
```



```
# generate embeddings
embedding = w2v.Word2Vec()
embedding.build_vocab(all_questions)
embedding.train(all_questions)
```



```
# Words model
words_model = Sequential()
words_model.add(Embedding(vocab_size, embedding_dim, weights=[embedding]))
words_model.add(LSTM(64, activation='softmax', return_sequences=True, dropout=0.5))
words_model.add(LSTM(64, activation='relu', return_sequences=False, dropout=0.5))
words_input = Input(shape=(max_len,))
processed_words = words_model(words_input)
```



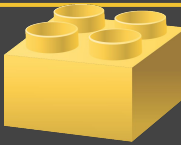
```
# Numbers model
nums_model = Sequential()
nums_model.add(Dense(3, input_dim=3))
nums_input = Input(shape=(3,))
processed_nums = nums_model(nums_input)

# Concatenate models
merged = keras.layers.concatenate([processed_nums, processed_words])

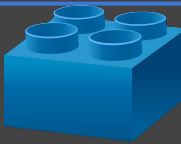
# add one dense layer to integrate the merging
hidden = Dense(32, activation='tanh')(merged)
hidden = Dropout(0.5)(hidden)

# finish with a dense layer
output = Dense(3)(hidden)
```

```
# generate embeddings
embedding = w2v.Word2Vec()
embedding.build_vocab(all_questions)
embedding.train(all_questions)
```

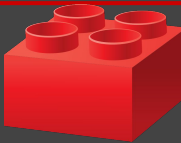


```
# Words model
words_model = Sequential()
words_model.add(Embedding(vocab_size, embedding_dim, weights=[embedding]))
words_model.add(LSTM(64, activation='softmax', return_sequences=True, dropout=0.5))
words_model.add(LSTM(64, activation='relu', return_sequences=False, dropout=0.5))
words_input = Input(shape=(max_len,))
processed_words = words_model(words_input)
```



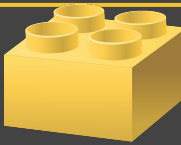
```
# Numbers model
nums_model = Sequential()
nums_model.add(Dense(3, input_dim=3))
nums_input = Input(shape=(3,))
processed_nums = nums_model(nums_input)

# Concatenate models
merged = keras.layers.concatenate([processed_nums, processed_words])
```

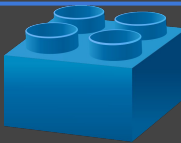


```
# add one dense layer to integrate the merging
hidden = Dense(32, activation='tanh')(merged)
hidden = Dropout(0.5)(hidden)
# finish with a dense layer
output = Dense(3)(hidden)
```

```
# generate embeddings
embedding = w2v.Word2Vec()
embedding.build_vocab(all_questions)
embedding.train(all_questions)
```

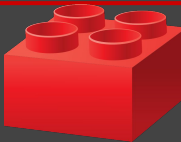


```
# Words model
words_model = Sequential()
words_model.add(Embedding(vocab_size, embedding_dim, weights=[embedding]))
words_model.add(LSTM(64, activation='softmax', return_sequences=True, dropout=0.5))
words_model.add(LSTM(64, activation='relu', return_sequences=False, dropout=0.5))
words_input = Input(shape=(max_len,))
processed_words = words_model(words_input)
```



```
# Numbers model
nums_model = Sequential()
nums_model.add(Dense(3, input_dim=3))
nums_input = Input(shape=(3,))
processed_nums = nums_model(nums_input)

# Concatenate models
merged = keras.layers.concatenate([processed_nums, processed_words])
```



```
# add one dense layer to integrate the merging
hidden = Dense(32, activation='tanh')(merged)
hidden = Dropout(0.5)(hidden)
# finish with a dense layer
output = Dense(3)(hidden)
```

Results

Input:

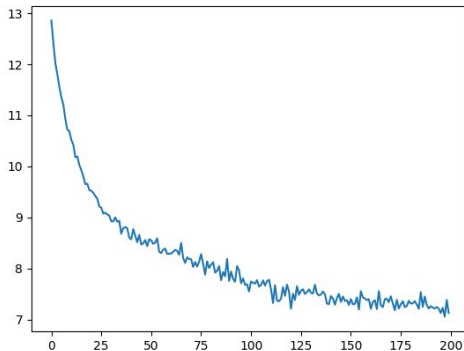
['You have 5 apples...',
'23 years from now...',
'Sean collected 47 stamps...',
'A cup of coffee costs 13.5\$...',
'Fourteen drinks were served ...']

Prediction:

$5 * x = 39 - 19$
 $2 * x = 23 - 16.8$
 $5 * x = 47 - 17$
 $2 * x = 13.5 - 3.5$
 $4 * x = 14 - 6$

Truth:

$5 * x = 39 - 19$
 $2 * x = 23 - 16.8$
 $5 * x = 47 - 17$
 $2 * x = 13.5 - 3.5$
 $-4 * x = 6 - 14$



PART I Recap



PART II - Get a Side Project!



PART II - Get a Side Project!

PART II - Get a Side Project!

“hello world”



PART II - Get a Side Project!

“hello world”



yes!



PART II - Get a Side Project!

“hello world”



yes!



publish???



PART II - Get a Side Project!

“hello world”



yes!



publish???



Take Home Message



DREAM BIG

Thank you :)

Questions?

DalyaG@gmail.com

TeMS -
Textual Math Solver

Simplisico

 seematics

VISIONARY
DEEP LEARNING

