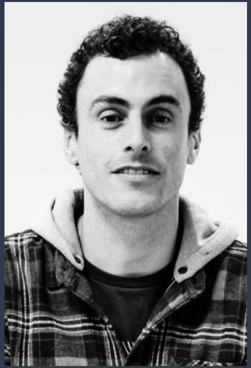
DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Slides available in repo!

Lab Time!
From LR to Transformers
Task: Sentiment analysis

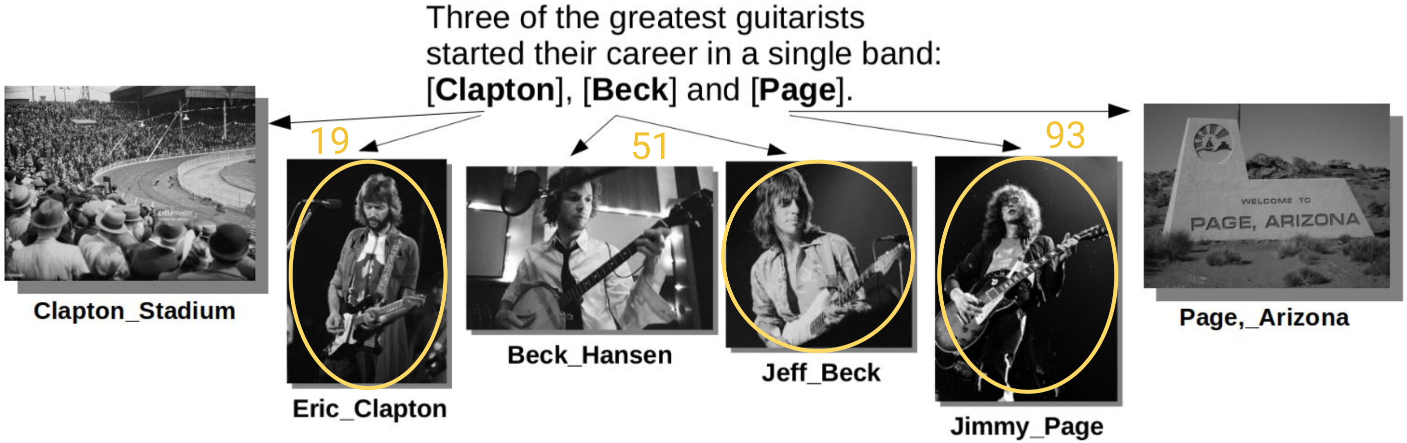
About myself



- PhD in Natural Language Processing in 2017 (EHU) Machine Learning! (old stuff!)
  - PGM and Generative Statistical Models
  - Personalized Page Rank
- post-Doctoral grant from EHU (Deep Learning)
- assistant professor in EHU
- research interest:
  - **Multilingual/Unsupervised Named Entity Disambiguation**
  - Sequence Labeling, **Language Modeling (Basque, Medical domain)**, QA, Semantic Textual Similarity, Information Extraction...
  - Deep Learning for NLP
  - **Brain Image Classification based on Transformers**
    - Dyslexia
    - Alzheimer
    - ...

About myself

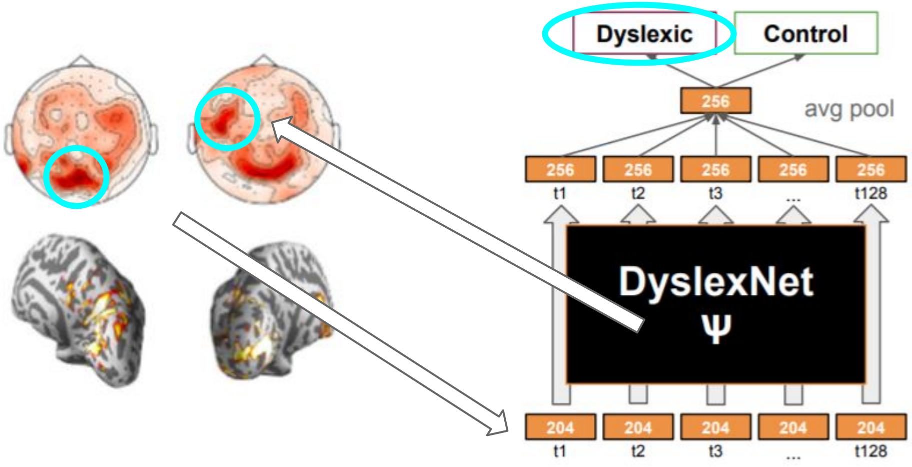
- Research interest:
  - **Multilingual/Unsupervised Named Entity Disambiguation**



Three of the greatest guitarists started their career in a single band: [**Clapton**], [**Beck**] and [**Page**].

About myself

- Research interest:
  - **Brain Image Classification based on Transformers**
    - Dyslexia
    - Alzheimer
    - …



46[sub] x 3k[instances] x **128[time] x 204[activations]**

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Slides available in repo!

TensorFlow!

Lab Time!
From LR to Transformers
Task: Sentiment analysis

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

- Rise your hand and ask your question
- Try not to use chat (only for you)
- Private room (share your screen...)
  - small talks…

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Lab1:
Warming Up
+ assignment
-> sol. next lab

Lab1: Warming Up

- Follow the instructions in 0.Instructions.docx to access the lab data.
- Goal: Ensure everithing is working fine.
  - Python (list comprehension)
  - Numpy (vector multiplications)
  - TensorFlow 2.0
  - Loading/Examining data from labs folder
    - Train/Dev/Test splits
  - 2 goals -> numpy and TF2.0
- Code in Jupiter Colab & self-explanatory labs.
- We recommend you the python/numpy tutorial and to take a look to the slides.

Please! play with the code!

Change numbers and print each step vector size!

Run code more than once!

Lab1: Warming Up

- Follow the instructions in 0.Instructions.docx to access the lab data.
- Goal: Ensure everithing is working fine.
  - Python
  - Numpy
  - TensorFlow 2.0
  - Loading/Examining data from labs folder
  - 2 assignments numpy and TF2.0
- Code in Jupiter Colab.
- We recommend you the python/numpy tutorial and to take a look to the slides.

$$\frac{1}{\sqrt{2\sigma^2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Today’s Goal!
Function Broadcasting!

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus
Hitz Zentroa - Ixa Taldea

em@il (subject: lab1 - name) me with the colab link from [share - get link] and grant permissions to view

Attendance certificate
Progress certificate (all labs, not assignments)

Good luck!

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus
Hitz Zentroa - Ixa Taldea

em@il (subject: lab1 - name) me with the colab link from [share - get link] and grant permissions to view

Attendance certificate
Progress certificate (all labs, not assignments)

lab1 until 18:30!

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Assignment1: Logistic Regression, SGD, and Regularization

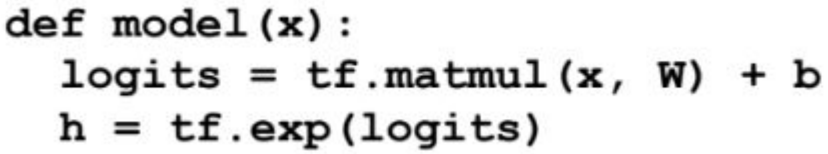
Assignment1: Logistic Regression, SGD, and Regularization

- Logistic Regression model for sentence classification using Tensorflow.
- Task: Sentiment analysis
- 5-way classification task into 2-way classification task (0→negative, 1→positive)
- Goals:
  - Pick an effective learning rate for SGD
  - Implement L2 regularization (see bellow)
  - Pick an effective L2 weight
  - Look at some learning curves (optional)
- Check:
  - BoW feature vector
  - Num of parameters?
  - Hyperparameters?
  - Norm and Exp
  - mini Batch
  - data spliting
  - ...

Assignment1: Logistic Regression, SGD, and Regularization

- Logistic Regression model for sentence classification using Tensorflow.

$$h = \exp(xW + b)$$



```
def model(x):
  logits = tf.matmul(x, W) + b
  h = tf.exp(logits)
```

- **Overfitting and regularization:** W can be very good for training, with enough layers and capacity the model can memorize the training data!
  - Generalize very poorly to test data (= the real world)
- First solution: add a regularizer to the loss function that avoids the model to fit the training data.
  - Squared L2 norm

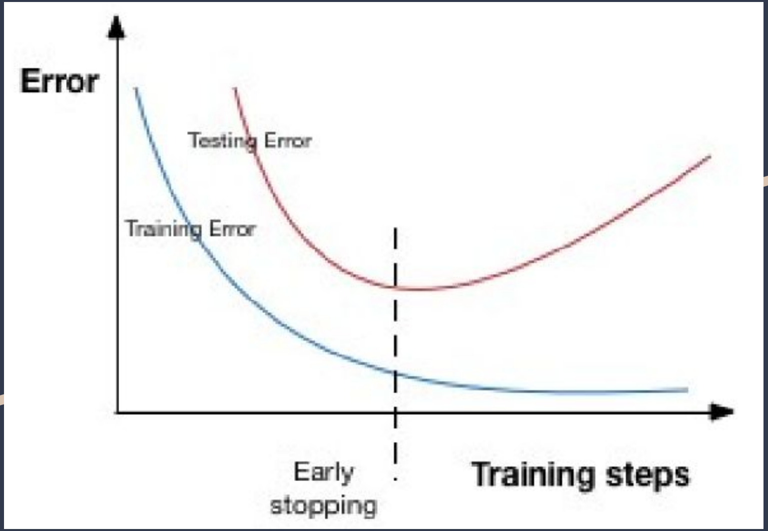
Assignment1: Logistic Regression, SGD, and Regularization

- Logistic Regression model for sentence classification using Tensorflow.
- **Overfitting and regularization:** W can be very good for training, with enough layers and capacity the model can memorize the training data!
  - Generalize very poorly to test data (= the real world)
- **First solution:** add a regularizer to the loss function that avoids the model to fit the training data.
  - Squared L2 norm

$$J_i(W) = -\log\left(\frac{\exp(W_{c_i}^T x)}{\sum_{c' \in C} \exp(W_{c'}^T x)}\right) \boxed{+\lambda \sum_k W_k^2}$$

Check the TF documentation! for sum and square

Assignment1: Logistic Regression, SGD, and Regularization



- Logistic Regression model for sentence classification using Tensorflow.
- **Overfitting and regularization:** W can be very good for training, with enough layers and capacity the model can memorize the training data!
  - Generalize very poorly to test data (= the real world)
- **First solution:** add l2 regularizer to the loss function that avoids the model to fit the training data.
- **Second solution:** Early stopping finishes training as soon as development error starts to increase
  - **Experimental setup:**
    - %80 train, %10 development, %10 test (blind!!)
  - **Model selection:**
    - best accuracy (lowest error) at development

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Good luck!
Sol. next Lab!

Don't send me the assignment!

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Lab Time!

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus
Hitz Zentroa - Ixa Taldea

em@il (subject: lab1 - name) me with the colab link from [share - get link] and grant permissions to view

Attendance certificate
Progress certificate (all labs, not assignments)

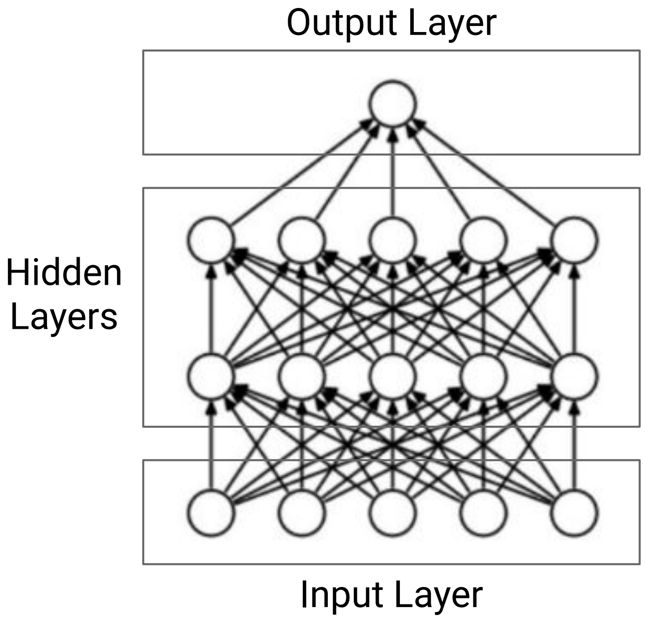
DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Lab2: MLP and Dropout

Lab2:
MLP and Dropout

- Modify a Logistic Regression to turn it into a MLP with two hidden layers (non-linearity).
  - 2 hidden layers = Deep Learning...
- It will **overfit** in training data!



Today's Goal!

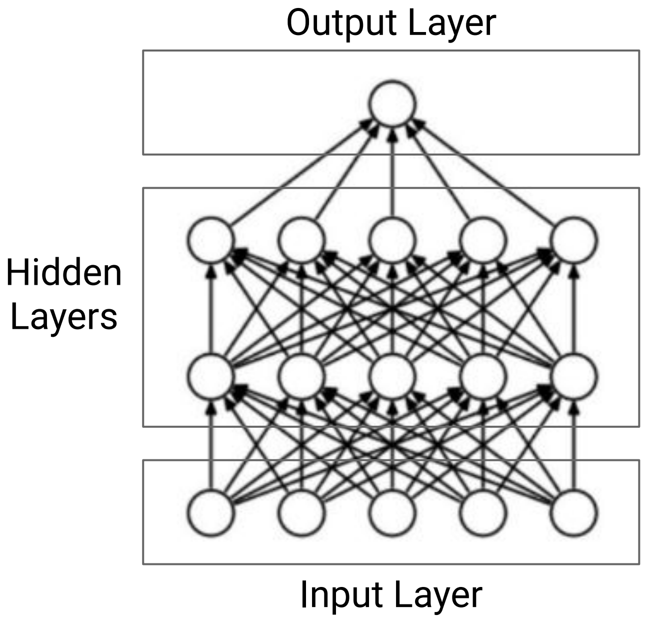
Lab2:
MLP and Dropout

$$y = softmax(h_1 W_2 + b_2)$$

$$h_1 = f(h_0 W_1 + b_1)$$

$$h_0 = f(x W_0 + b_0)$$

- Modify a Logistic Regression to turn it into a MLP with two hidden layers (non-linearity).
  - 2 hidden layers = Deep Learning...
- It will **overfit** in training data!



Today's Goal!

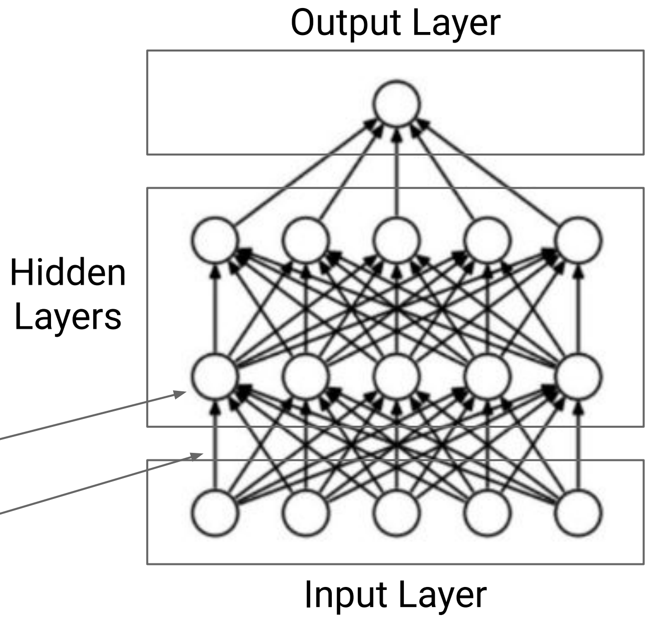
Lab2:
MLP and Dropout

$$y = softmax(h_1 W_2 + b_2)$$

$$h_1 = f(h_0 W_1 + b_1)$$

$$h_0 = f(x W_0 + b_0)$$

- Modify a Logistic Regression to turn it into a MLP with two hidden layers (non-linearity).
    - 2 hidden layers = Deep Learning...
- It will **overfit** in training data!



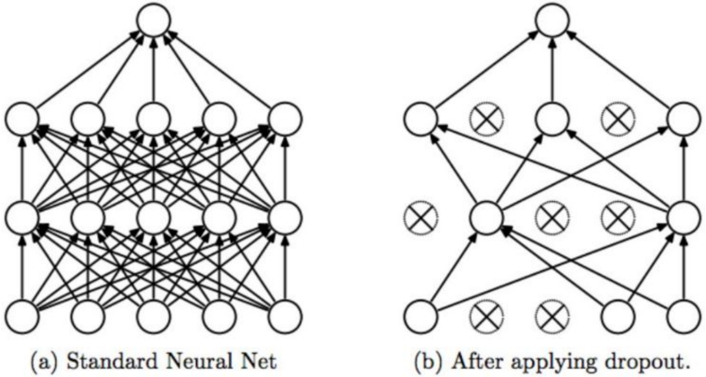
Sentence Representation (word order?)

Word Embeddings

Today's Goal!

Lab2:
MLP and Dropout

- Modify a Logistic Regression to turn it into a MLP with two hidden layers (non-linearity).
- It will overfit in training data!
- Add **Dropout** regularization. Hint: after non-linearity and only during training.



Source: “Dropout: a simple way to prevent neural networks from overfitting”, JMLR 2014

Lab2:
MLP and Dropout

- Modify a Logistic Regression to turn it into a MLP with two hidden layers (non-linearity).
- It will overfit in training data!
- Add **Dropout** regularization. Hint: after non-linearity and only during training.
- More parameters to tune!
  - Hidden layer number and size. The more the better? not always...
  - Dropout Probability.
  - Should we add dropout in the input layer?
  - Early stop?
    - Check where should we stop training and save the best model

Lab2:
MLP and Dropout

- Modify a Logistic Regression to turn it into a MLP with two hidden layers (non-linearity).
- It will overfit in training data!
- Add **Dropout** regularization. Hint: after non-linearity and only during training.
- More parameters to tune!
  - Hidden layer number and size. The more the better? not always...
  - Dropout Probability.
  - Should we add dropout in the input layer?
  - Early stop?

Be careful with dropout at test time

Be careful with dimensionalities

Be careful with the initialization!
Gradient Vanishing!

Lab2:
MLP and Dropout

- Modify a Logistic Regression to turn it into a MLP with two hidden layers (non-linearity).
- It will overfit in training data!
- Add **Dropout** regularization. Hint: after non-linearity and only during training.
- More parameters to tune!
    - Hidden layer number and size. The more the better? not always...
    - Dropout Probability.
    - Should we add dropout in the input layer?
    - Early stop?

Shared task! Share your **dev**/test results and best model

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Good luck!
Sol. next Lab!

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Lab Time!

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus
Hitz Zentroa - Ixa Taldea

em@il (subject: lab1 - name) me with the colab link from [share - get link] and grant permissions to view

Attendance certificate
Progress certificate (all labs, not assignments)

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

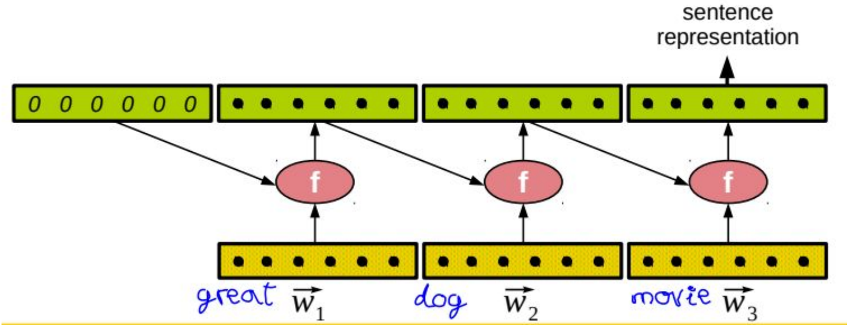
Lab3: RNN based classifier

Lab3:
RNN based classifier

Hints:

- Use tf.nn.embedding_lookup
- Print shapes tf.shape()
- Be patient...
- We already add l2 regularization...
- Improving dev at 100~200 ep
  - ~72% acc.

- Modify a Logistic Regresion to turn it into a RNN sentence classifier.
- Slower that MLP, and not very good... yet!
  - Don't try to improve results...
- Define **RNN** parameters (Recurrent layer)
- Define a recurrent step
  - Embedding_lookup
  - Concat
  - matmul & Tanh!
- Unroll to obtain sentence representation!
  - Loop over the sequence_lenght and use x_slices
  - Be careful with shapes! tf.reshape!
  - Perform an step
- You can add an MLP on top!



DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Assignment2: Word Embeddings

Assignment2: Word Embeddings!

- Cosine Similarity: Similar and related Words
- Semantic orientation
- Analogy
  - man -> king
  - woman -> ???
- Visualization tools
- Have a good time after RNN lab!

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Good luck!
Sol. next Lab!

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Lab Time!

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus
Hitz Zentroa - Ixa Taldea

em@il (subject: lab1 - name) me with the colab link from [share - get link]

Attendance certificate

Progress certificate (all labs, not assignments, send it before showing the solution)

DL4nlp - Deep Learning for Natural Language Processing

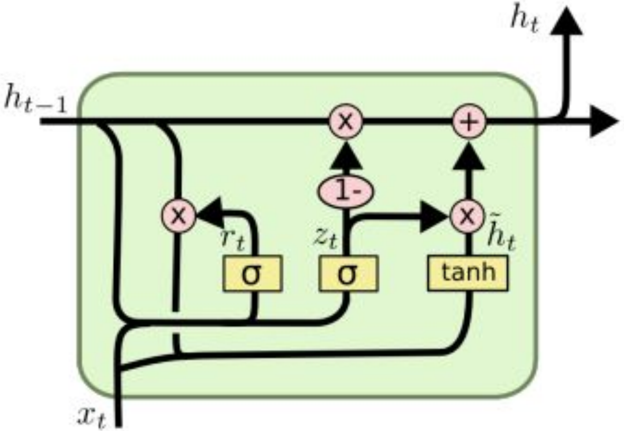
Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Lab4: GRU based classifier

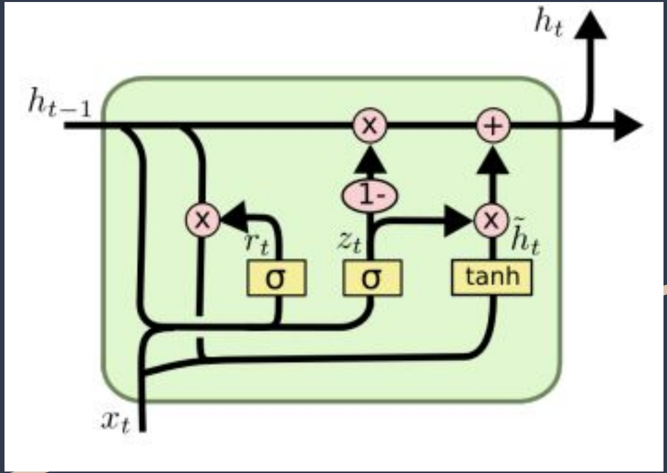
Lab4:
GRU based classifier

Kyunghyun Cho (2014)

- Modify a RNN to GRU.



Lab4:
GRU based classifier



- Modify a RNN to GRU.
  - Add aditional parameters
  - Modify the step function
  - Modidy L2 regularization
  - Wait 100~150 epochs… be patient
    - dev acc ~80% !!!
  - Plot dev/train acc
    - Check when do we need to early stop
  - Code LSTM or Stack GRU layers

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$ Update Gate

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$ Reset Gate

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

History

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

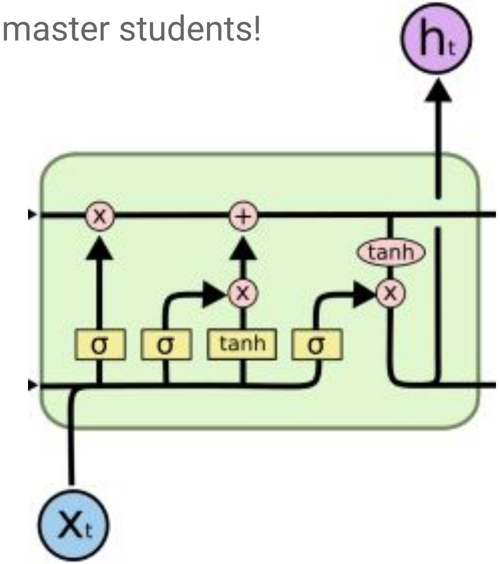
DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Assignment3: Language Modelling with LSTM networks

Assignment3: LSTM based classifier

- Modify a RNN to LSTM, perform **language modeling task** and answer some questions.
- Language Modeling, new task, new cost function…
- Same as GPT3 but using LSTMs an at low scale…
- Language Generation: Samples from a trained models should have “coherent” portions.
- Note that this is for master students!



DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Good luck!
Sol. next Lab!

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Lab Time!

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

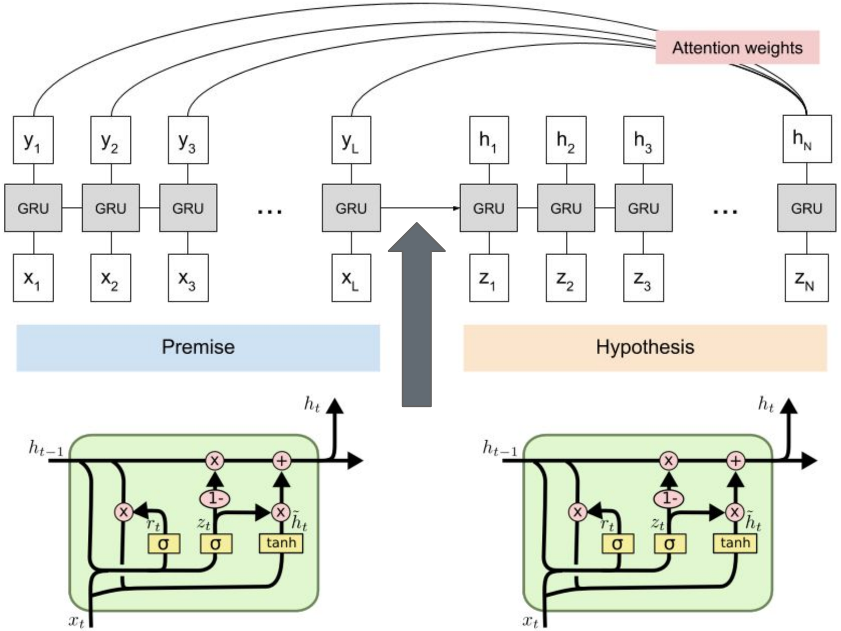
Lab5: Attention models

Lab5: Attention models

- Attention model for NLI based on Rocktäschel's model. (Read the paper...)
- Natural Language Inference (NLI). Given a pair of premise and hypothesis texts, the task is to classify them into three categories: entailment, contradiction, and neutral.
    - A man inspects the uniform of a figure in some East Asian country. | **contradiction** | The man is sleeping.
    - A soccer game with multiple males playing. | **entailment** | Some men are playing a sport.
    - An older and younger man smiling. | **neutral** | Two men are smiling and laughing at the cats playing on the floor.
- This is deep learning! (a scientific paper reimplementation)

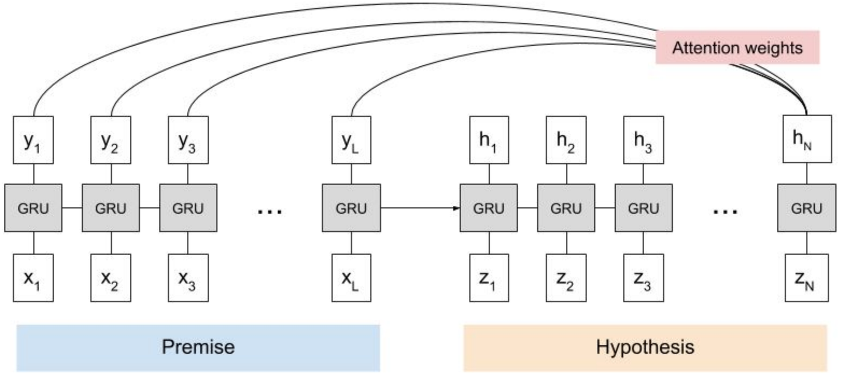
Lab5: Attention models

- Attention model for NLI based on Rocktäschel's model.
  - 2xGru encoders + attention (no bias)



Lab5: Attention models

- Attention model for NLI based on Rocktäschel's model.
  - 2xGru encoders + attention (no bias)



$$\mathbf{M} = \tanh(\mathbf{W}^y\mathbf{Y} + \mathbf{W}^h\mathbf{h}_N \otimes \mathbf{e}_L)$$

$$\alpha = \text{softmax}(\mathbf{w}^T\mathbf{M})$$

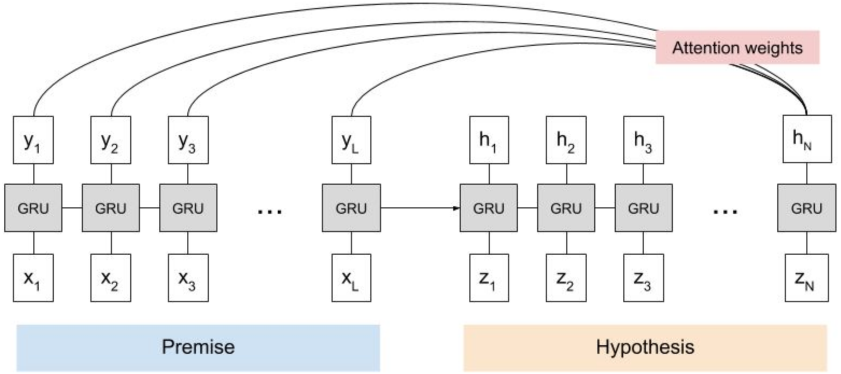
Goal!

$$\mathbf{r} = \mathbf{Y}\alpha^T$$

$$\mathbf{h}^* = \tanh(\mathbf{W}^p\mathbf{r} + \mathbf{W}^x\mathbf{h}_N)$$

Lab5: Attention models

- Attention model for NLI based on Rocktäschel's model.
  - 2xGru encoders + attention (no bias)



$$\mathbf{M} = \tanh(\mathbf{W}^y\mathbf{Y} + \mathbf{W}^h\mathbf{h}_N \otimes \cancel{\mathbf{e}_L})$$

$$\alpha = \mathrm{sof}\cancel{\mathrm{t}}\mathrm{max}(\mathbf{w}^{\cancel{T}}\mathbf{M})$$

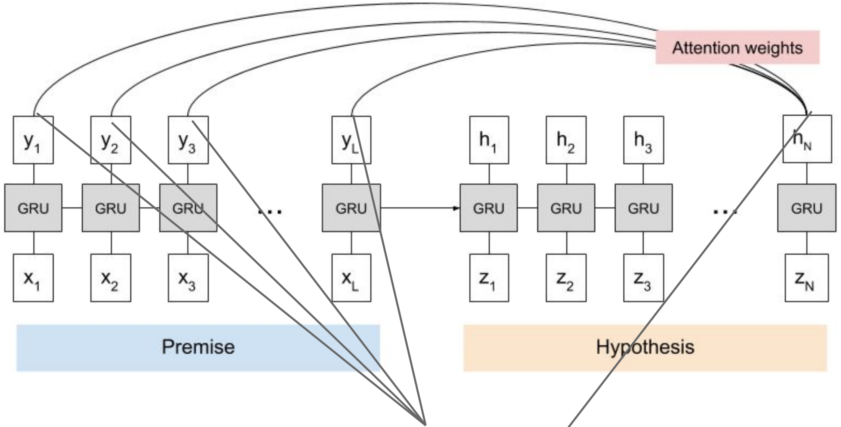
$$\mathbf{r} = \mathbf{Y}\alpha^T$$

$$\mathbf{h}^* = \tanh(\mathbf{W}^p\mathbf{r} + \mathbf{W}^x\mathbf{h}_N)$$

Hint! iterate over Y

Lab5: Attention models

- Attention model for NLI based on Rocktäschel's model.
  - 2xGru encoders + attention (no bias)



$$\mathbf{M} = \tanh(\mathbf{W}^y\mathbf{Y} + \mathbf{W}^h\mathbf{h}_N \otimes \cancel{\mathbf{e}_L})$$

$$\alpha = \mathrm{sof}\cancel{\mathrm{t}}\mathrm{max}(\mathbf{w}^{\cancel{T}}\mathbf{M})$$

$$\mathbf{r} = \mathbf{Y}\alpha^T$$

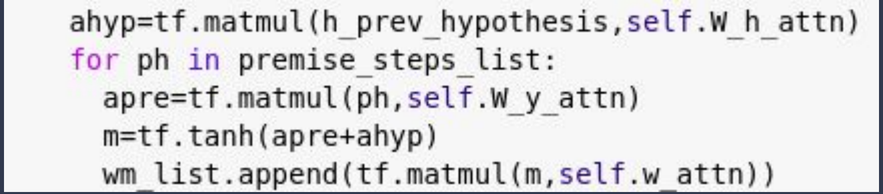
avg. representation

Hint! iterate over Y

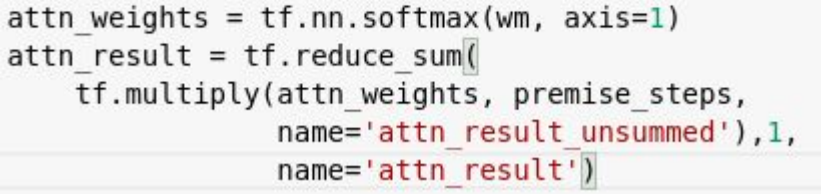
$$\mathbf{h}^* = \tanh(\mathbf{W}^p\mathbf{r} + \mathbf{W}^x\mathbf{h}_N) \longrightarrow$$

???

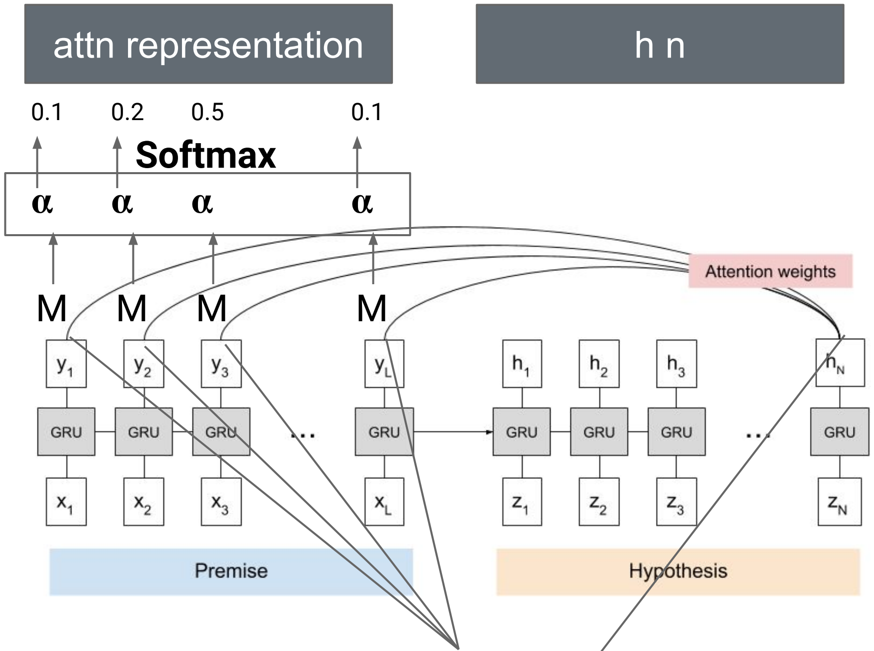
Lab5: Attention models solution



```
  ahyp=tf.matmul(h_prev_hypothesis,self.W_h_attn)
  for ph in premise_steps_list:
    apre=tf.matmul(ph,self.W_y_attn)
    m=tf.tanh(apre+ahyp)
    wm_list.append(tf.matmul(m,self.w_attn))
```



```
attn_weights = tf.nn.softmax(wm, axis=1)
attn_result = tf.reduce_sum(
    tf.multiply(attn_weights, premise_steps,
                name='attn_result_unsummed'),1,
                name='attn_result')
```



$$\mathbf{M} = \tanh(\mathbf{W}^y\mathbf{Y} + \mathbf{W}^h\mathbf{h}_N \otimes \xcancel{\mathbf{e}}_L)$$

$$\alpha = \mathrm{sof}\xcancel{\mathrm{t}}\mathrm{max}(\mathbf{w}^{\xcancel{T}}\mathbf{M})$$

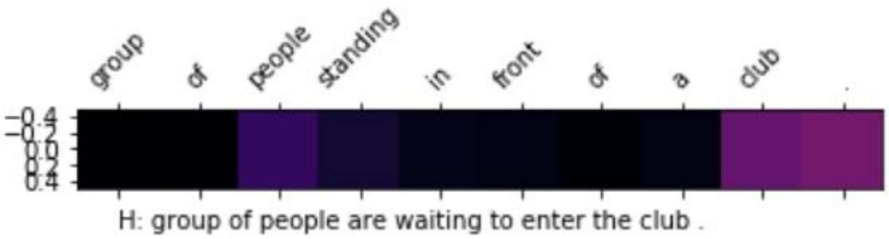
$$\mathbf{r} = \mathbf{Y}\alpha^T$$ avg. representation

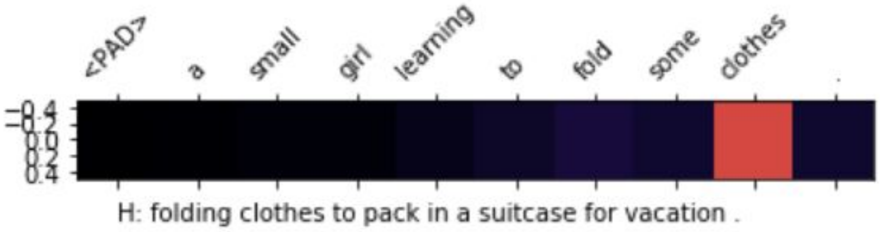
Hint! iterate over Y

$$\mathbf{h}^* = \tanh(\mathbf{W}^p\mathbf{r} + \mathbf{W}^x\mathbf{h}_N) \rightarrow ???$$

Lab5: Attention models

- Attention model for NLI based on Rocktäschel's model.
  - 2xGru encoders + attention (no bias)
  - Fixed and freezed embeddings (from Glove)
  - GPU! (enable in runtime) still slow training…
  - Don't expect high accuracy
- Once you are done
  - Plot attention weights





Lab5: Attention models

- Attention model for NLI based on Rocktäschel's model.
  - 2xGru encoders + attention (no bias)
  - Fixed and freezed embeddings (from Glove)
  - GPU!
  - This is deep learning!
- Once you are done
  - Plot attention weights
  - Back propagate through embeddings (more parameters...)
  - Increase hidden size, embedding size (dowload Glove embeddings)...
  - Try to increase model accuracy! (However, it takes too long...)

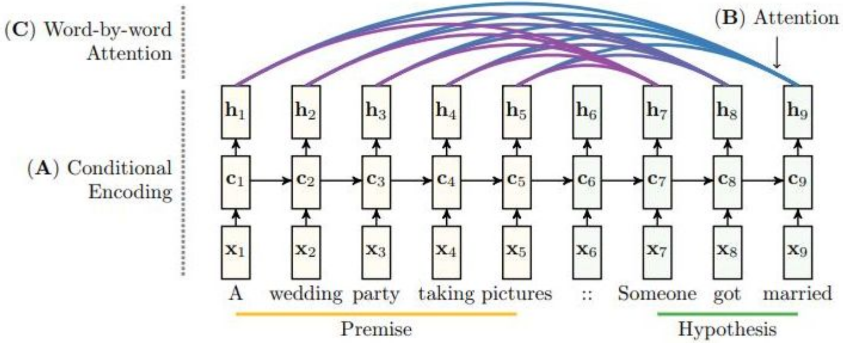
DL4nlp - Deep Learning for Natural Language Processing

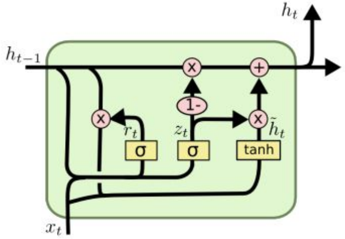
Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Assignment4: Word by Word attention

Assignment4: Word by Word attention

- Attention model for NLI based on Rocktäschel's model.
    - 2xGru encoders + Word by Word attention


(C) Word-by-word Attention
(B) Attention
(A) Conditional Encoding
$\mathbf{h}_1$ $\mathbf{h}_2$ $\mathbf{h}_3$ $\mathbf{h}_4$ $\mathbf{h}_5$ $\mathbf{h}_6$ $\mathbf{h}_7$ $\mathbf{h}_8$ $\mathbf{h}_9$
$\mathbf{c}_1$ $\mathbf{c}_2$ $\mathbf{c}_3$ $\mathbf{c}_4$ $\mathbf{c}_5$ $\mathbf{c}_6$ $\mathbf{c}_7$ $\mathbf{c}_8$ $\mathbf{c}_9$
$\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_3$ $\mathbf{x}_4$ $\mathbf{x}_5$ $\mathbf{x}_6$ $\mathbf{x}_7$ $\mathbf{x}_8$ $\mathbf{x}_9$
A wedding party taking pictures :: Someone got married
Premise
Hypothesis



DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Good luck!
Sol. next Lab!

DL4nlp - Deep Learning for Natural Language Processing

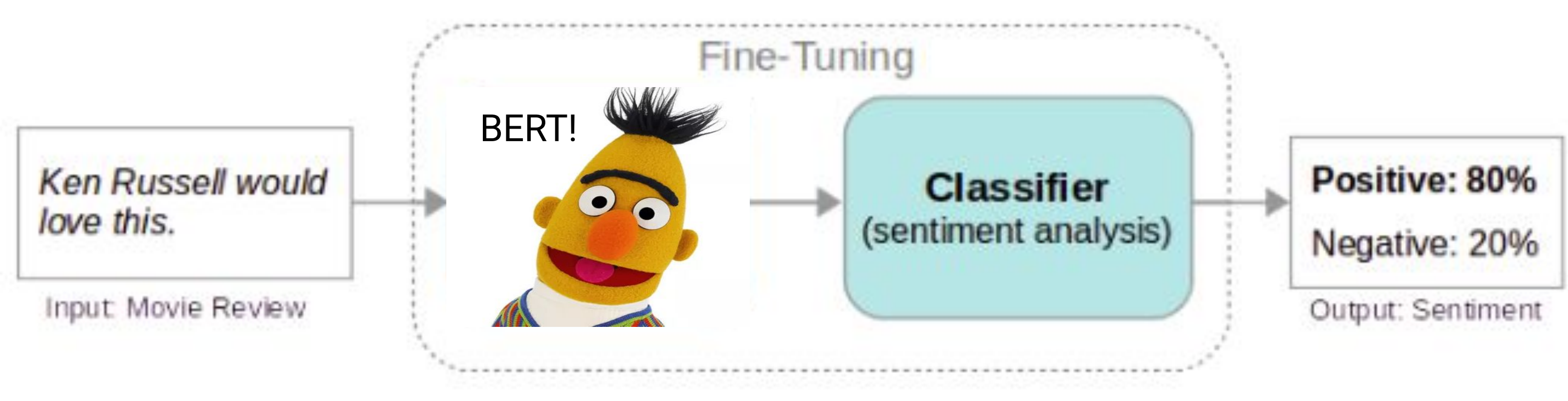
Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Lab5: Transformers with TensorFlow

Lab5: Transformers with TensorFlow

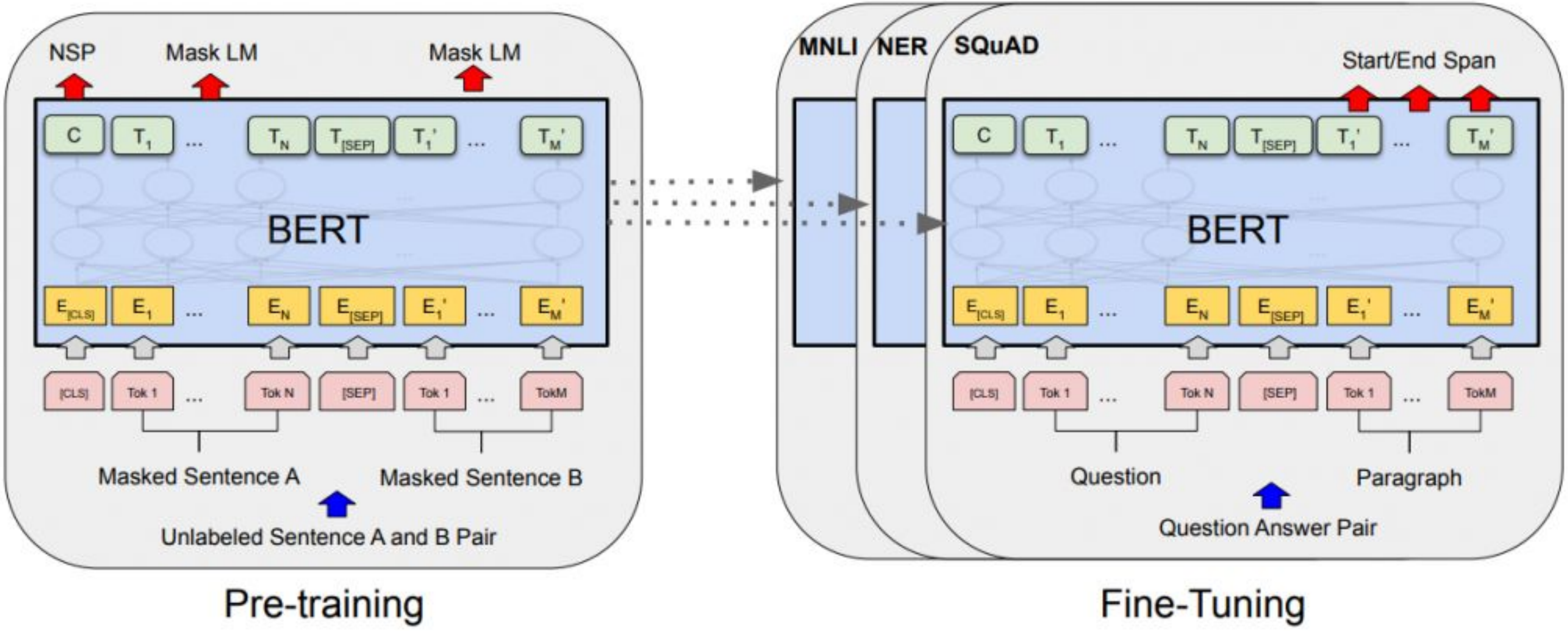
- Fine-tune a Transformer based pre-trained language models.
  - 3 or 4 epochs!
  - we introduce:
    - keras
    - huggingface transformers

Black box!



Lab5: Transformers with TensorFlow

- **BERT** pretraining: **MLM & NSP** and **Fine-tuning**



Lab5: Transformers with TensorFlow

- Fine-tune a Transformer based pre-trained language models.
  - 2, 3 or 4 epochs!
- More concretely, in this lab session will learn the following:
  - Deploy and fine-tune transformers (**BERT**).
  - Transfer learning!
  - Preprocessing data for transformers archicture (word piece tokenization)
  - **soa** Transformer-based classifier
    - High test Acc -> **0.90!!**
  - Finetune for NLI:
    - **Hint:** Zip and List Comprehension + batch_encode_plus

Lab5: Transformers with TensorFlow

Once you are done:

- Try training with **XLMR (crossling model)**
- Run in other datasets or tasks
- Test with sentences in other languages

- Fine-tune a Transformer based pre-trained language models.
  - 2, 3 or 4 epochs!
- More concretely, in this lab session will learn the following:
  - Deploy and fine-tune transformers (**BERT**).
  - Transfer learning!
  - Preprocessing data for transformers archicture (word piece tokenization)
  - **soa** Transformer-based classifier
    - High test Acc -> **0.90!!**
  - Finetune for NLI:
    - **Hint:** Zip and List Comprehension + batch_encode_plus

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Good luck :D

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Last Lab :(

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Lab6: Image Captioning

Lab6: Image Captioning

- Given an image like the example below, our goal is to generate a caption such as "**a surfer riding on a wave**". (cherry picking! :P)

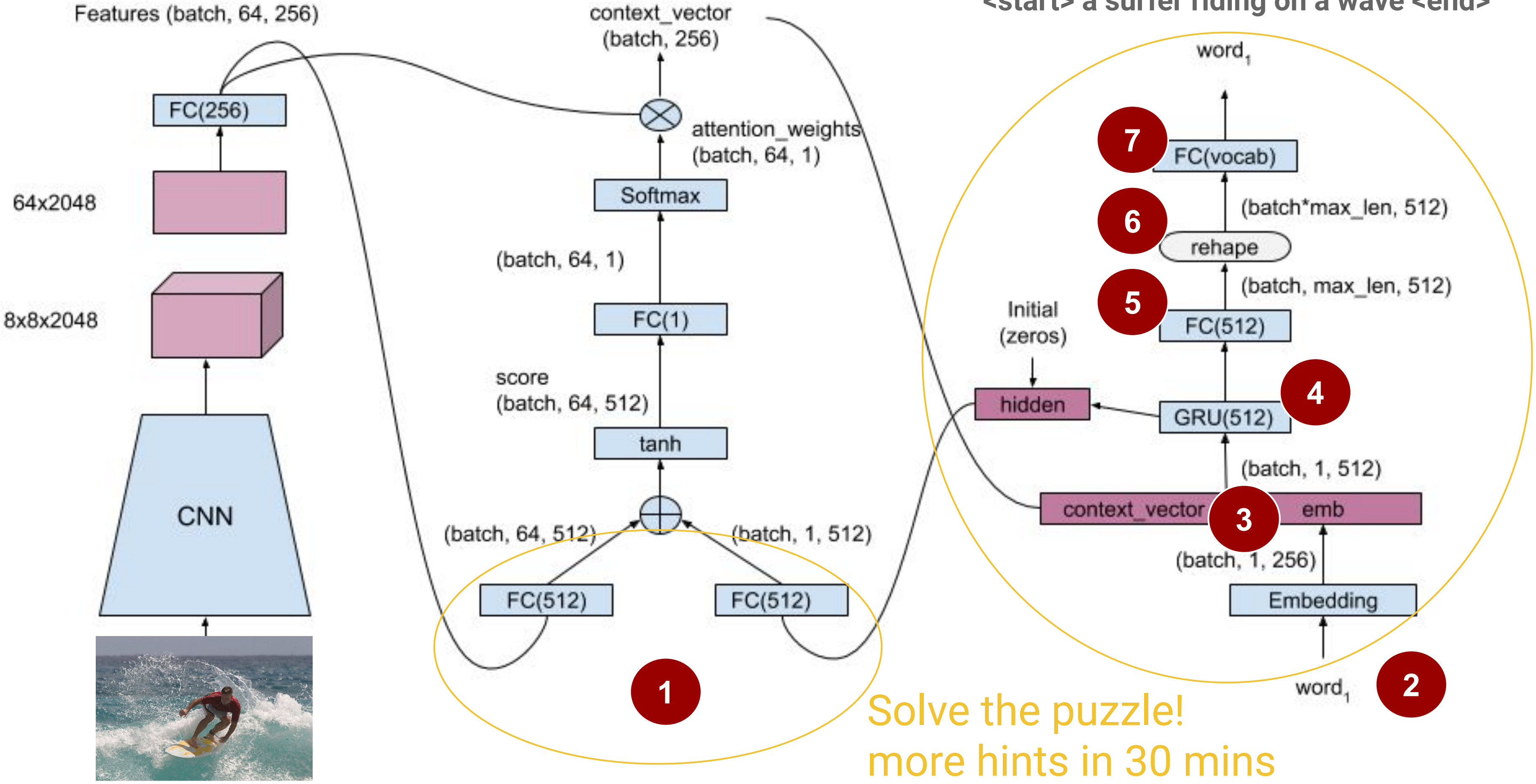
- Toy examples = bad captions!
- Attention maps!
- Try on your on images

Lab6: Image Captioning

- Keras
- CNN image encoder (pre-trained!)
- GRU-RNN text decoder
- Attention (image & text)

- Multimodal task
- Transfer learning (InceptionV3 for image classification)
- Image and text preprocessing cells…
- Checkpoints!

“<start> a surfer riding on a wave <end>"



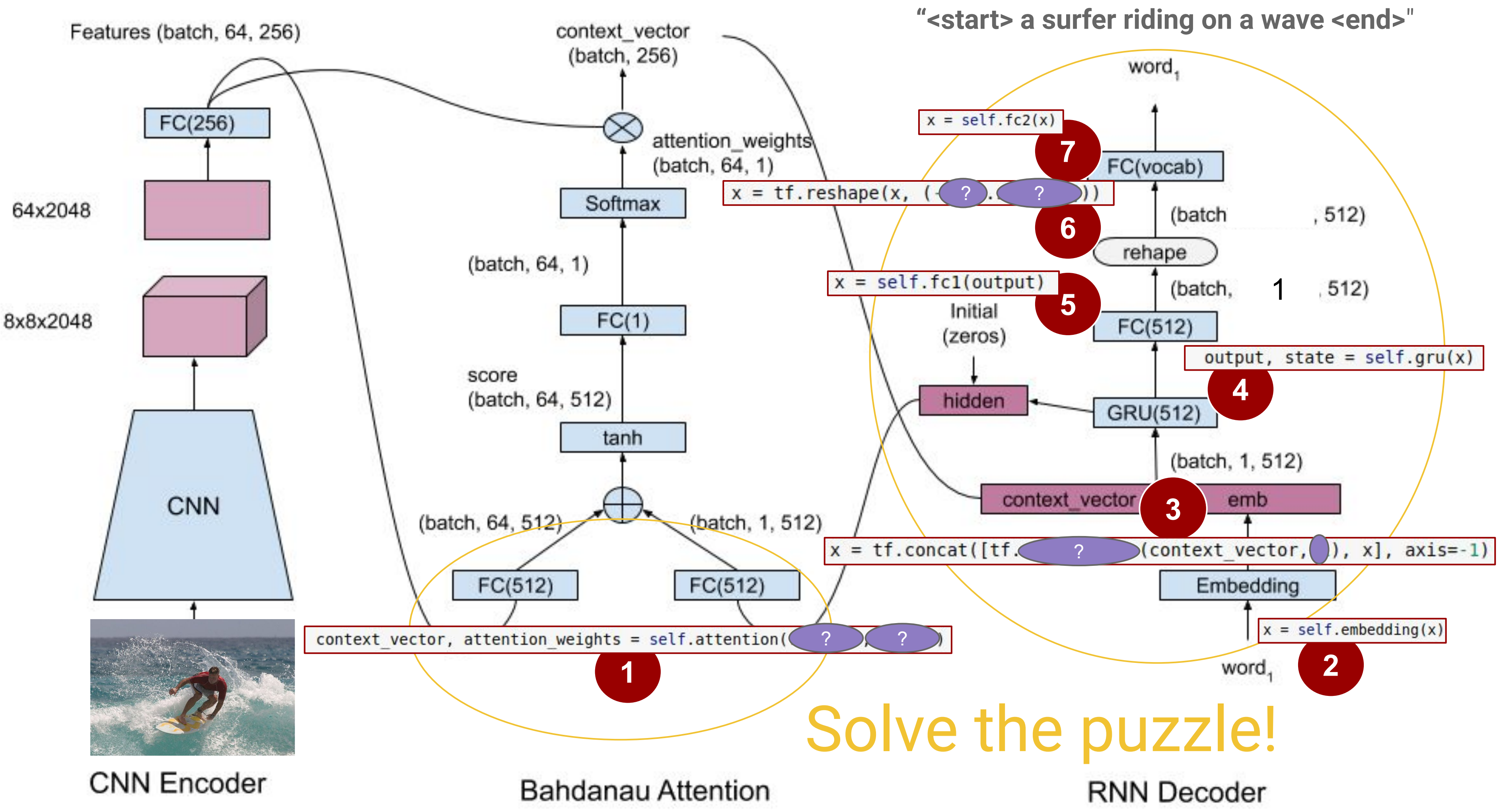
Solve the puzzle!
more hints in 30 mins

CNN Encoder

Bahdanau Attention

RNN Decoder

“<start> a surfer riding on a wave <end>"



```
context_vector, attention_weights = self.attention( ? , ? )
x = self.embedding(x)
x = tf.concat([tf. ? (context_vector, ? ), x], axis=-1)
output, state = self.gru(x)
x = self.fc1(output)
x = tf.reshape(x, ( ? , ? ))
x = self.fc2(x)
```

Solve the puzzle!

DL4nlp - Deep Learning for Natural Language Processing

Ander Barrena Madinabeitia
ander.barrena@ehu.eus - @4nderB
Hitz Zentroa - Ixa Taldea

Good luck!