**Google Colab disconnect code**

var startClickConnect = function startClickConnect(){

var clickConnect = function clickConnect(){

console.log("Connnect Clicked - Start");

document.querySelector("#top-toolbar > colab-connect-button").shadowRoot.querySelector("#connect").click();

console.log("Connnect Clicked - End");

};

var intervalId = setInterval(clickConnect, 60000);

var stopClickConnectHandler = function stopClickConnect() {

console.log("Connnect Clicked Stopped - Start");

clearInterval(intervalId);

console.log("Connnect Clicked Stopped - End");

};

return stopClickConnectHandler;

};

var stopClickConnect = startClickConnect();

**Dataset**

* **Yelp Dataset** (<https://www.yelp.com/dataset>)
  + ~ 8M reviews, ~200k businesses, ~1.9M users
* **Official Documentation:** <https://www.yelp.com/dataset/documentation/main>

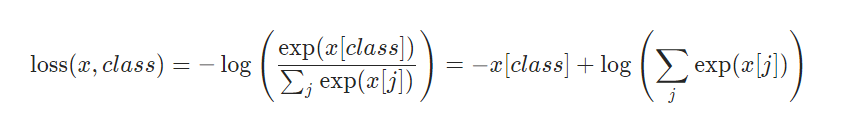
**Dataset Selection**

* Businesses
  + **Input:** ~200k businesses
  + US Businesses (~153k businesses, ~73% of all businesses)
  + Open businesses (~125k businesses, ~81% of all businesses)
  + Restaurant Business in tags (use Top 100 restaurant categories)
    - 'Restaurants', 'Food', 'Bars', 'American (Traditional)', 'Fast Food', 'Sandwiches', 'Coffee & Tea', 'Breakfast & Brunch', 'Pizza', 'Burgers', 'Mexican', 'American, …….
  + Businesses with > 30 reviews
  + **Output:** ~18k businesses
* Reviews Dataset
  + **Input:** ~8M reviews
  + Selects business present in previous filtering
  + Users with > 30 reviews available
  + Businesses with > 30 reviews available
  + **Output:** ~609k reviews
* Subsampling for Modelling
  + Uneven Distribution of Reviews
    - 5\*: 35.5%, 4\*: 35.5%, 3\*: 17.1%, 2\*: 7.5%, 1\*: 4.5%
  + Randomly Subsample 20k reviews from each category (100k reviews)
  + 90k for Training, 10k for Test

**Procedure**

* Data Filtering & Sampling
* Preprocess Reviews Data
  + Remove Punctuation, Links URL, New Line character, Replace Multiple spaces
  + Lower case text
  + Tried to do Stemmer + Spell Correction (BUT took too long)
* Tokenizer for BERT
  + Used HuggingFace BERT Tokenizer (‘bert-base-uncased’)
  + Add [CLS] in front of reviews, and [SEP] at the end of reviews
  + Tokenize and Trim reviews to only 350 tokens
  + Create Segment Mask, and Attention Mask
* BERT Modelling
  + Used Dataloader (Batch Size = 16)
  + BERT-base model uncased fine-tuning
  + CrossEntropy Loss
  + Need labels to be from 0, 1, 2, n\_classes – 1
  + Train for 3 epochs of 90k data each
* Measure Metrics: Accuracy, Sentiment, Precision, Recall, F1-score (Macro-avg)
* Pick best model and do prediction
* Create front end and back end
* Deployment

**BERT Model Notes**

* Use bert-base uncase. Previously tried bert-base uncased but performed worse
* Tokenizer convert words to ids -> Trim to 350 tokens
* Input to BERT Model
* Take [CLS] word embedding -> Summary of all sequence sentence
* Stack multiple Dense, Tanh, and dropout layer to produce better performance
  + Initialize all Dense layers with Xavier Initialization
* Then do Cross Entropy Loss
* Use Cross Entropy Loss (classification with 5 classes)
  + For Loss, need to have labels from 0, 1, 2, 3, num\_classes - 1
* Use Adam Optimizer, **Learning Rate = 5e-5, # Epochs = 3 of 90k each**
* Evaluate model every 30k steps
* **To Save Tokenizer/Base Model to local:** Model.save\_pretrained(), Tokenizer.save\_pretrained()
* Model Architecture

Cross Entropy

* Bert Hidden Size = 768
* Bert Output (batch\_size x seq\_len x 768)
* Take the [CLS] token embedding (batch\_size x 768)
* Dense Layer1 = 768 output (batch\_size x 768)
* Tanh1 (batch\_size x 768)
* Dropout1 p=0.1 (batch\_size x 768)
* Dense Layer2 = 256 output (batch\_size x 256)
* Tanh2 (batch\_size x 256)
* Dropout2 p=0.1 (batch\_size x 256)
* Dense3 = 128 output (batch\_size x 128)
* Tanh3 (batch\_size x 128)
* Dense4 = 5 output (batch\_size x 5)
* Cross Entropy (batch\_size x 5)

Dense 4

Tanh 3

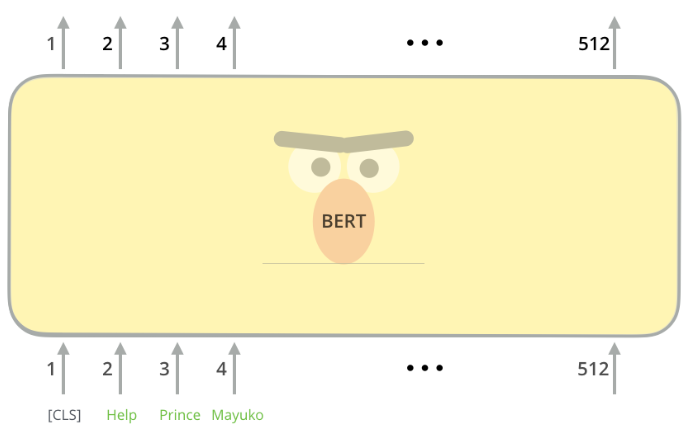
Dense 3

Dropout 1,2

2x

Tanh 1,2

Dense 1,2



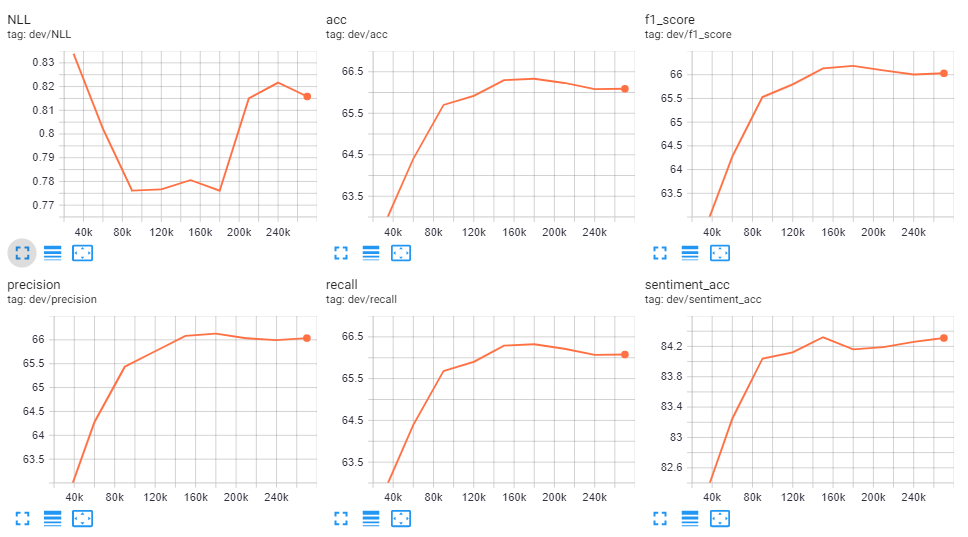
350

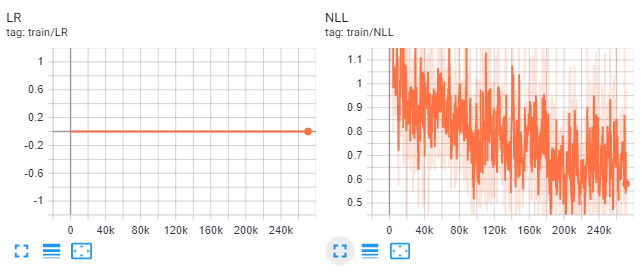
350

[SEP]

**Metrics**

* Accuracy (How many correct)
* Sentiment
  + 4\*/5\* = Positive
  + 3\* = Neutral
  + 1\*/2\* = Negative
  + How many predictions match?
* Precision (Macro Avg)
* Recall/Sensitivity (Macro Avg)
* F1-Score (Macro Avg)
* **Macro Avg:** Calculate metrics for each label, and find their unweighted mean. This does not take label imbalance into account.

**Results**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Iteration | Loss | Acc | Sentiment Acc | Precision | Recall | F1 |
| 30k | 0.83 | 62.75 | 82.11 | 62.49 | 62.74 | 62.58 |
| 60k | 0.80 | 64.40 | 83.25 | 64.27 | 64.39 | 64.28 |
| 90k | 0.78 | 65.70 | 84.04 | 65.44 | 65.68 | 65.53 |
| 120k | 0.78 | 65.92 | 84.12 | 65.76 | 65.90 | 65.80 |
| 150k | 0.78 | 66.30 | 84.32 | 66.08 | 66.29 | 66.14 |
| 180k | 0.78 | 66.33 | 84.16 | 66.13 | 66.32 | 66.19 |
| 210k | 0.82 | 66.23 | 84.19 | 66.03 | 66.22 | 66.09 |
| 240k | 0.82 | 66.08 | 84.26 | 65.99 | 66.07 | 66.00 |
| 270k | 0.82 | 66.09 | 84.31 | 66.04 | 66.08 | 66.03 |

Training Time: ~ 3 hours using Google Colab Tesla P100 GPU

Best Model: 180k Iteration

After 180k iterations, model starts to overfit to the training data.

**Pytorch Save Model**

* When training default uses torch.save(model\_save\_dict(), PATH)
  + Can be used to continue Training
  + However, we need to have model class defined again
  + <https://pytorch.org/tutorials/beginner/saving_loading_models.html>
* Save Jitted Model: Torch. jit.save (torch.jit.trace(model, (inputs\_to\_model), PATH)
  + Saves entire model but cannot be used for any other architectures
  + Used for fast loading of model
  + <https://discuss.pytorch.org/t/how-to-load-using-torch-load-without-source-class-using-which-model-was-created/75302>

**Selected Examples (Good)**

* **Example1:** 
  + **Reviews:** I absolutely loved the nachos here. I consider myself a nacho connoisseur. These are some of the best I've ever had. Big enough for two people. Interesting variety of nachos available. I love this place so much if possible id open a franchise where I live, in Virginia Beach. A MUST VISIT for us from now on when we go on our yearly Vegas trip!!!!!!
  + **Answer:** 5
  + **True Sentiment**: Positive
  + **Prediction:** 5
  + **Prediction Sentiment**: Positive
* **Example2:**
  + **Reviews:** Not too bad! Been to this location several times, still have not loved it yet. The antipasta platter was ok. The mozzarella was good; the brushette had too much pesto; and calamari was not crispy enough. The chicken marsala was ok too. I tasted more of the grill of the chicken than the marsala sauce. The mash potatoes were good. I should have stuck to my usual pasta carrabba. The service is always excellent!
  + **Answer:** 3
  + **True Sentiment**: Neutral
  + **Prediction:** 3
  + **Prediction Sentiment**: Neutral
* **Example3:**
  + **Reviews:** I honestly do not understand peoples infatuation with this place. The fries are terrible and the burgers are barely edible. I have tried several In-N-Out Burgers to make a fair assessment, and they're all nasty.
  + **Answer:** 1
  + **True Sentiment**: Negative
  + **Prediction:** 1
  + **Prediction Sentiment**: Negative
* **Example4:**
  + **Reviews:** If I could give White Castle a negative star I would. I can't believe this place exists. We had a gift card and wanted to try it out. We ordered 40 mini sliders, fries and onion rings. The meat didn't even look or taste like meat, it was horrible. Everything was soggy and not eatable it felt like I was eating something foreign. The best part about the meal was the Powerade. Very upsetting.
  + **Answer:** 1
  + **True Sentiment**: Negative
  + **Prediction:** 1
  + **Prediction Sentiment**: Negative

**Selected Examples (Bad)**

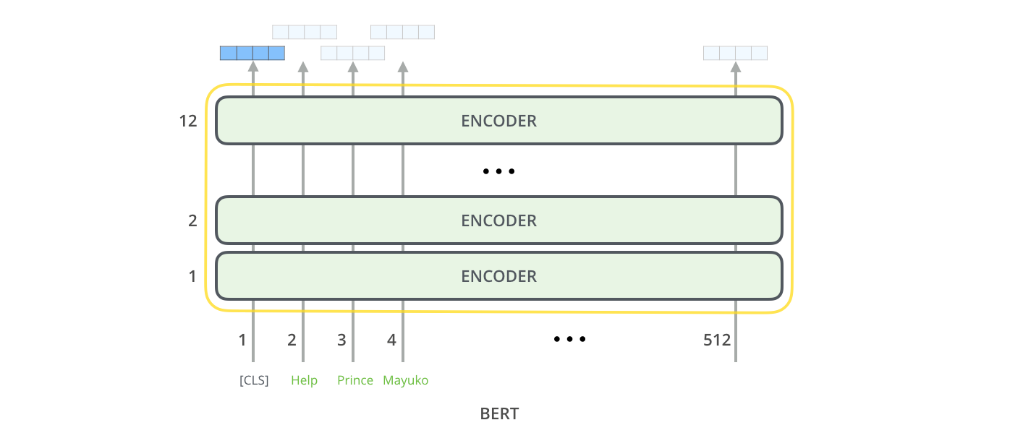
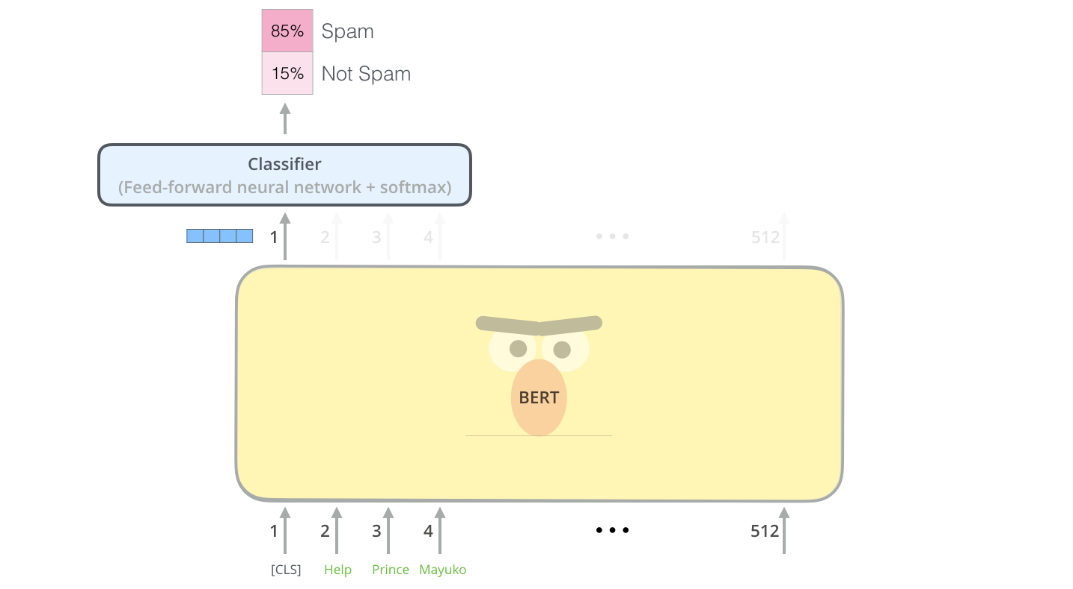
* **Example4:**
  + **Reviews:** Got a group of friends together for a Saturday night and thought HBJ's would be a great place to cut loose. The restaurant is set up with long tables all picnic style but it worked for us. Our waitresses were AMAZING and deserve a 5 star rating, kept separate tabs for all of us, handled a group of 12 with ease. The food was so so but the place is meant to be a country bar so I can understand food isn't their priority.

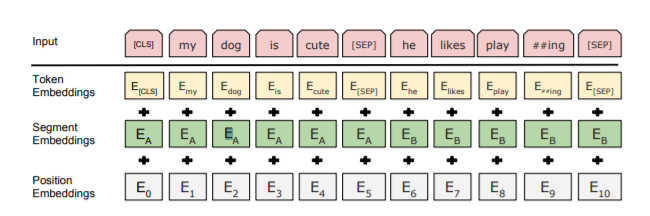
My low review is based on the owner. After dinner, we all wanted to mingle a bit and asked if we could take the large high top table right behind us. I asked if we could keep part of our dinner table just in case someone wanted to sit and our waitress said ok. Five minutes later an old lady came over and asked if we were keeping our dining table. I said we didn't need the entire table but wanted to keep some chairs in case we wanted to sit. The old lady slapped me on the back and told me she was the owner and was going to "kick me out" for asking to keep our table. Then she abruptly walked away. I was very confused but a few moments later she went up to another member of our group and threatened to kick them out too! Turns out she wanted to seat a group of her regulars at our table (even though I had called ahead and made a reservation).

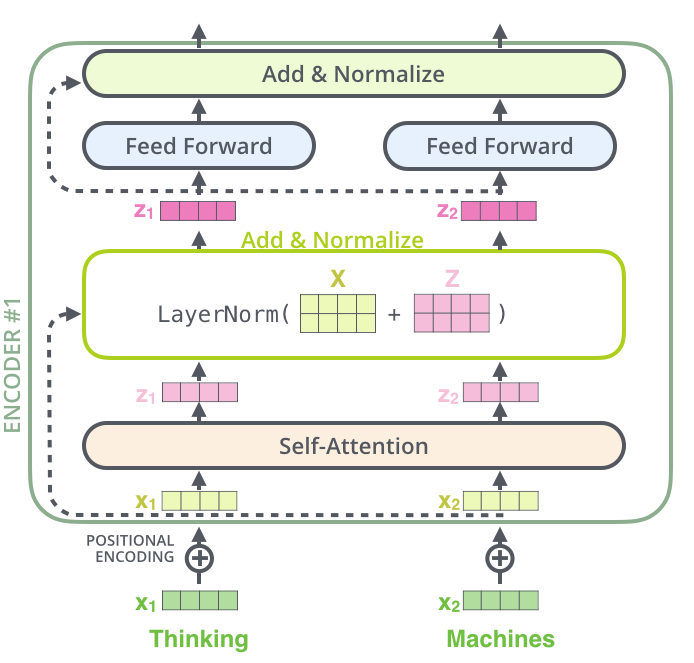
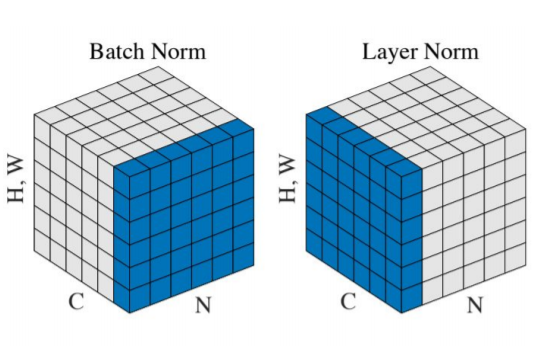
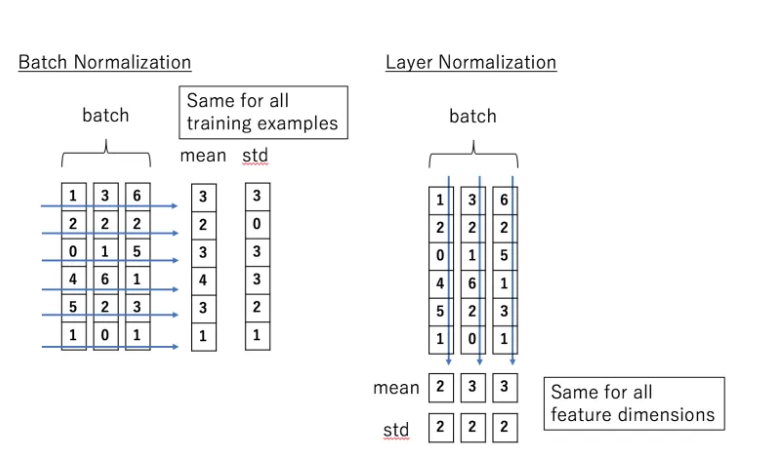
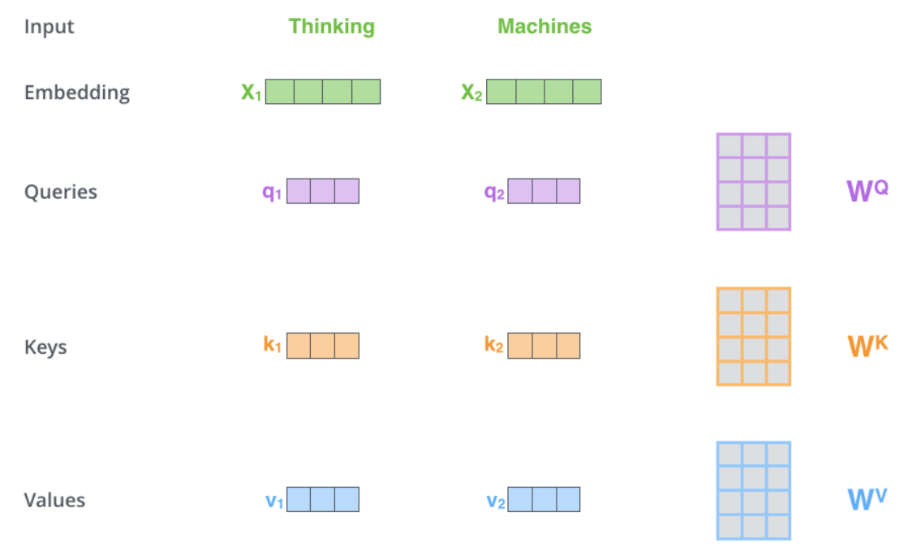
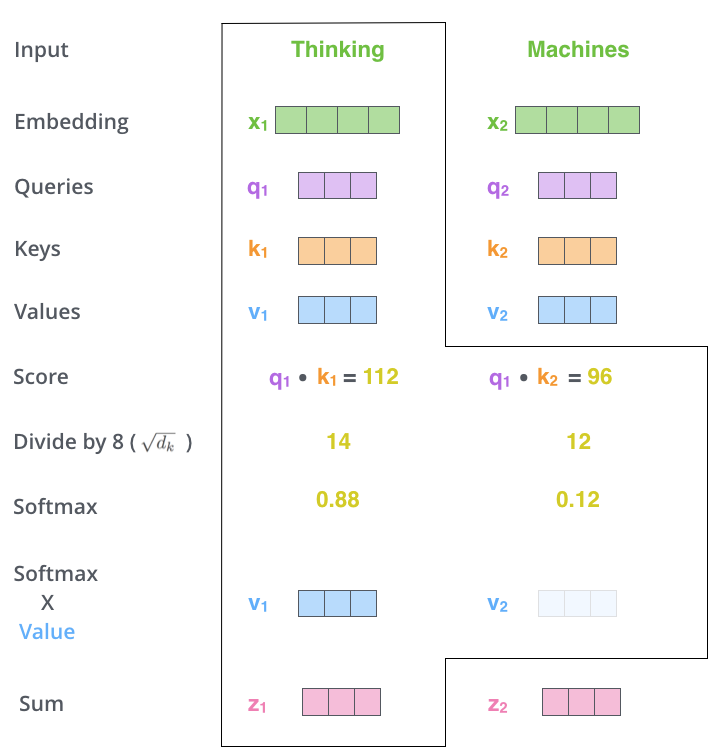
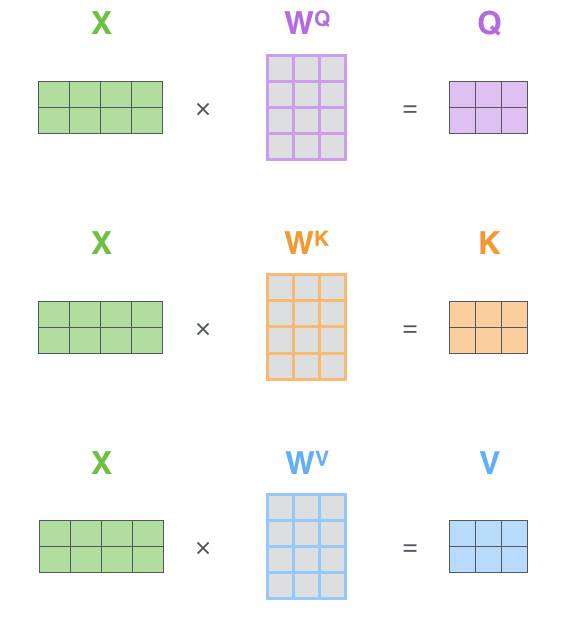
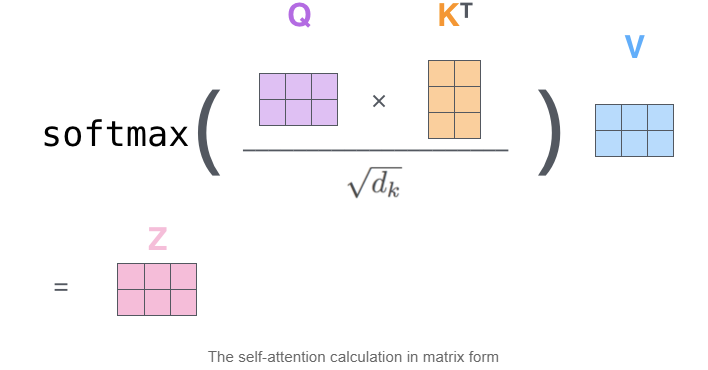
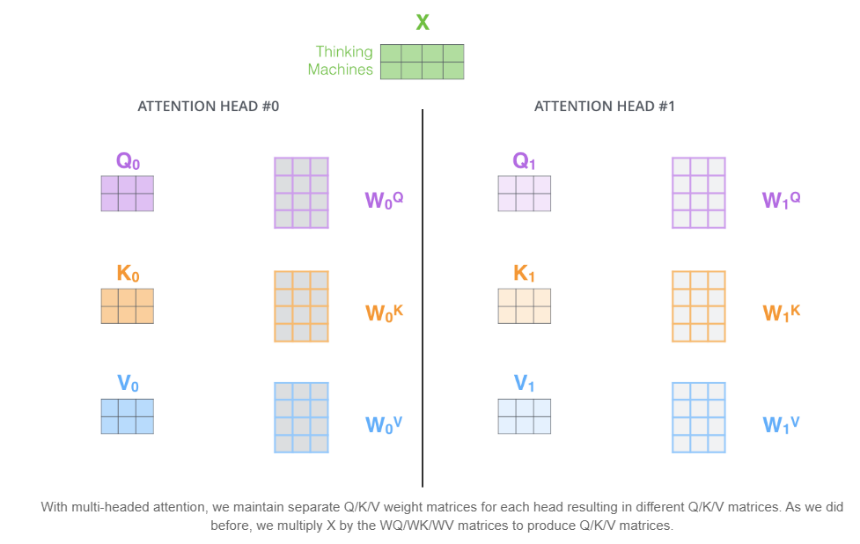
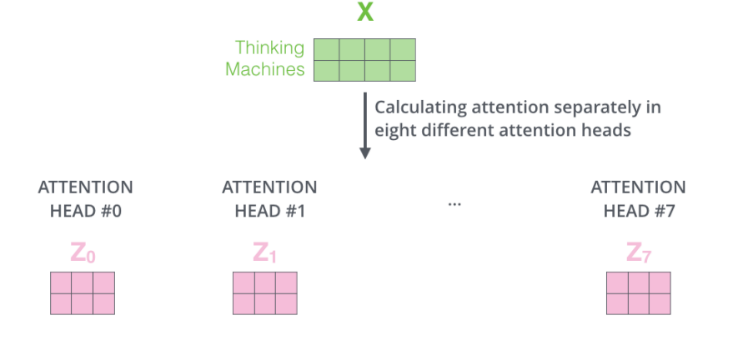
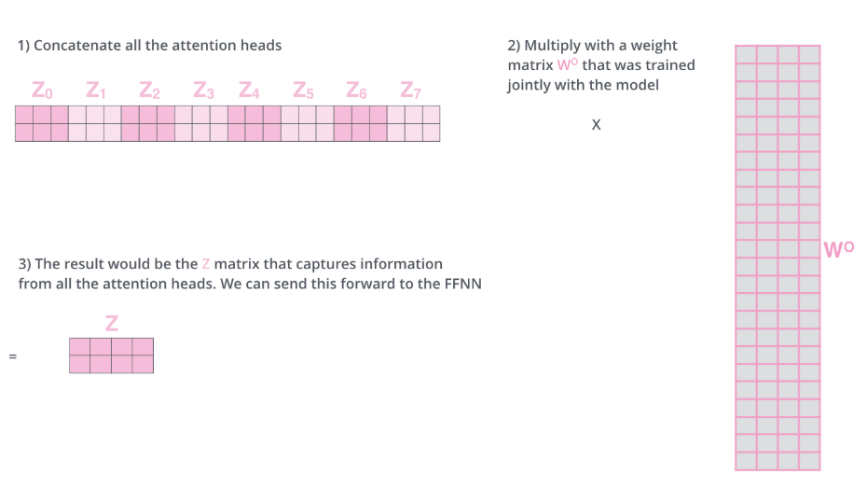
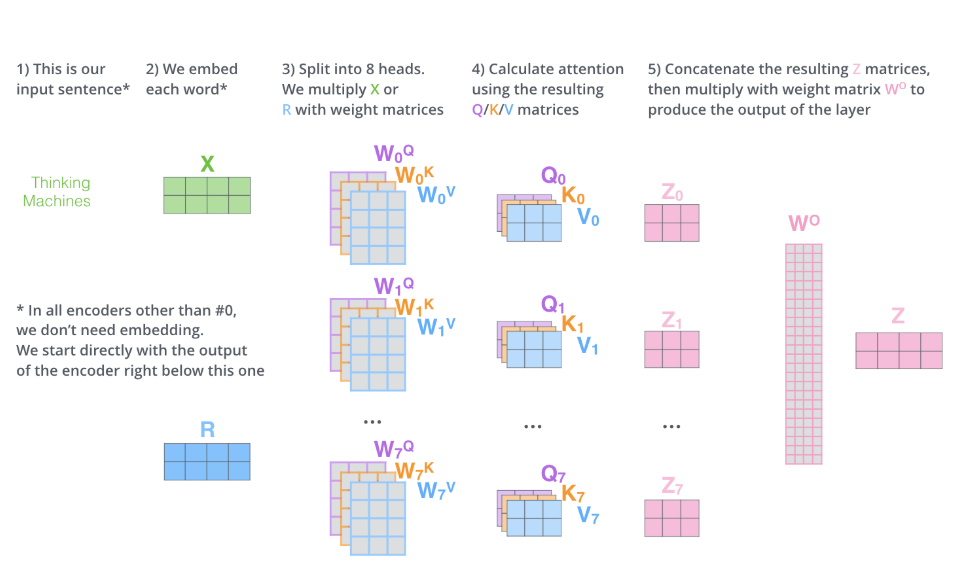
It was very clear she didn't want our business so we paid, tipped our server well and left. Probably won't go back either.

* + **Answer:** 1
  + **True Sentiment**: Negative
  + **Prediction:** 2
  + **Prediction Sentiment**: Negative
* **Example5:**
  + **Reviews:** We got there for an early dinner. Place didn't look that busy when we arrived. They took about five minutes to great and another five to sit us. I was not impressed by the way place look. Floors were dirty with food. But then I saw waiter cleaning tabla and dumping crumbs on floor. After we sat down waiter left and didn't come back to take our drink order for a long time. We almost got up and left because of how long they took to take our orders. We had the al Pastor mahi fish and mole tacos. They were super good!! I also had a michelada and it was delicious!! Food wise I give them Five stars. But service and cleanliness I give them two stars.  
    Next time I'll give the one in Glendale a chance. Hoping food is as good as here but with better service and a more clean environment.
  + **Answer:** 4
  + **True Sentiment**: Positive
  + **Prediction:** 2
  + **Prediction Sentiment**: Negative
* **Example6:**
  + **Reviews:** We tried the corned beef sandwich. I'm not the biggest fan of corned beef, but when I get a hankering for it, I need the real thing. The sandwich is pretty, with swirled pumpernickel bread and cheddar, but the corned beef appears to be the kind that comes in slices or a pack rather than the brisket we're used to. Plus, they fried the meat! We'll search some more for REAL corned beef.
  + **Answer:** 2
  + **True Sentiment**: Negative
  + **Prediction:** 3
  + **Prediction Sentiment**: Neutral

**BERT Model Theory**

* Bert is trained on 2 tasks
  + Semi-Supervised Learning
    - Masked Language Modelling on Wikipedia, and books
    - Ex: I bought a \_\_\_\_ of milk (Answer: gallon)
  + Supervised Learning
    - Spam vs Non-spam
    - Sentiment Analysis
    - Next Sentence Prediction
      * Ex: Bob drank a gallon of milk. Stock A went up 100% yesterday (Answer: False)
* BERT Base: 12 layers of encoder stack (attention-heads)
* Output = (seq\_len x 768), where 768 is hidden size of Bert-base model
* Visuals:
  + BERT for Word Embeddings
  + BERT for Classification
  + Word embedding of [CLS] is used to feed to classifier
* Since in this task we are interested in classifying for YELP ratings (1,2,3,4,5), we will only use [CLS] token embedding.
* What are the embeddings?
  + The embeddings consist of a combination of Token Embeddings (from tokenizer), Segment Embeddings (Sentence 1/Sentence 2 pair), and position Embeddings.
  + All sentence in this task will have [0] \* max\_len segment embeddings as there is only one sentence NOT pairs!



* What are the encoder blocks consist of?
  + Input: Word Embeddings
  + Contents: Self-Attention Layer + Layer Norm + Feed-Forward Neural Network
  + What is Layer Norm?
    - Normalize input per data point. Based on Addition of X and Z vectors.
    - More on X and Z vectors Later.
* **What is the Self-Attention Layer?**
  + **Terminology**
    - Input -> Embedding. For each word, we create **Query Vector, Key Vector, and Value Vector**
    - These 3 vectors are created by Multiplying Embedding Matrix with WQ, WK, WV.
    - The dimensions of the Q, K, V vectors are smaller (Ex: 64) compared to Original (Ex: 512)
  + **Calculate Attention Scores**
    - Calculate Score by dot product of q and k vectors
      * Q1 dot k1, Q1 dot k2, Q1 dot k3, …., Qn dot k1, Qn dot k2, …., Qn dot kn
    - Then dividing by square root of output dimensions (Ex: sqrt (64) = 8)
    - Apply softmax layer and multiply by value vector
    - Softmax scores represent determines how much each word will be expressed at this position. Clearly the word at this position will have the highest softmax score, but sometimes it’s useful to attend to another word that is relevant to the current word.
    - Multiply Softmax score by value vector
    - Sum the weighted value vectors, Z1 = V1 + V2 + ….. Vn
  + **Matrix Method of Calculating Self-Attention Scores (Same as previously but in matrix format)**
    - **Matrix Multiply for Q, K, V vectors**
    - Word embeddings are stacked horizontally to get seq\_len x Embedding size
    - WQ, WK, WV size: Embedding\_size x Hidden\_size
    - Q, K, V size: Seq\_len x Hidden\_size
    - **SoftMax Layer and multiply by V vector**
    - Q x K^T size: Seq\_len x Seq\_len
    - Softmax( Q x K^T) x V = Seq\_len x Hidden\_size
    - Z size = Seq\_len x Hidden\_size
* **Multi Head Attention**
  + Expands model ability to refer to other words when calculating attention
  + Transformers utilizes 8 Attention heads
  + Multi Head Attention Calculation
  + First calculate the 8 attention heads with W0Q, W0K, W0V, W1Q, W1K, W1V, ….
  + Then we have 8 attention heads (Z0, Z1, Z2, …, Z7) -> Concatenate and multiply by Wo (weights trained on model size: 8 \* hidden\_size x input to Feed Forward layer)
  + We can feed the result to Feed Forward Layer
* **Final Visualization of Multi Head Attention in Encoder Block**
  + Visual
  + Then we can take the Z matrix and pass it to Feed Forward Layer

**BERT REFERENCES**

* <https://arxiv.org/pdf/1810.04805.pdf> (Paper)
* <http://jalammar.github.io/illustrated-bert/>
* <https://jalammar.github.io/illustrated-transformer/>

**Fine-Tune BERT for Question Answering**

* Input: [CLS] Question [SEP] Answer [SEP]
* Learn weights S and E
* Take only last output layer of BERT Model (T) shape = seq\_len x 768
* Matrix multiply T and S -> Softmax
* Learn using Cross Entropy Loss (Log Softmax + Negative Log Likelihood)
* Learn rate = 5e-5
* # epochs = 3

**References**

* <https://www.ahmedbesbes.com/blog/end-to-end-machine-learning>
* <https://signal.onepointltd.com/post/102gc83/sentiment-analysis-with-yelp-and-bert>
* <https://github.com/onepointconsulting/yelp_bert/blob/master/bert_training.ipynb>
* <https://chatbotslife.com/predicting-yelp-reviews-using-bert-81c583f15340>
* <http://jalammar.github.io/illustrated-bert/>