Decrypting Deep Learning

Deep Learning steps are same as ML

Deep Learning uses Neural Networks

Machine Learning Building Steps

- 1) Problem Identification
- 2) Data Collection
- 3) Data Preparation
- 4) Model Selection
- 5) Model Training/ 6)Hyperparameter Tuning
- 7) Model Evaluation
- Repeating Steps 5-7

Deep Learning Applications: NLP and Computer Vision

Deep Learning Steps

- 1) Problem Identification
- 2) Data Collection + Data Augmentation
- 3) Data Preparation
- 4) Model Selection
- 5) Model Training/ 6)Hyperparameter Tuning + Feature Engineering
- 7) Model Optimization
- Model Evaluation

ML Step 1 : Problem Identification

- Classify Toxic Comments
  - Multilabel Text Classification: “Hello” vs “#$%% off”
  - Toxic, Severely Toxic, Identity-Hate, Threat, Obscene, Insult

- Classifying Animal pictures
  - Multilabel Image Classification



Deep Learning Toolkits

- Python, Jupyter Noteboks

- Text Comments: Pytorch Lightning
- Images: Keras, Tensorflow

ML Step 2: Data Collection

- Text Comments: Kaggle Competition

https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

- Images: Local Directory

Data Augmentation

Increasing your dataset size through common methods

Data Augmentation

- Text Comments: Translating Language from English to French and back to English
- This ends up changing the word makeup of sentence, while keeping meaning

- Images: Rotating Image, Transposing image
- Increases our dataset size with images look like taken from different angle

ML Step 3: Data Preparation

- Text Comments: Pytorch Dataset, DataLoader

- Image: Keras method to load from directory

ML Step 3: Data Preparation

- Splitting up data for Training, Testing, Evaluation

- Text Comment: SKLearn Train_test_split can be used with Pytorch

- Image: Same Keras method that loads data splits it

ML Step 4: Model selection

- Text Classification: BERT Tokenizer + Simple Linear NN

- 2D CNN

ML Step 5/6: Model Training/Hyperparameter Tuning + Feature Space Engineering

- Train Neural Network Models over the course of multiple epochs
- Input Data is split and fed in by batches.
- Input is turned into numeric format
- Inputs are fed through different layers in a forward pass
- Loss function informs the model how well it's doing
- Based on loss, Backpropagation recomputes NN weights and Biases
- Added benefit: Weights and Biases that determine Features are also optimized

Feature Space Engineering

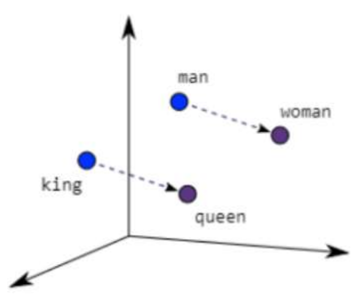
- Neural Networks need numeric inputs

- Words are not Numbers
- Images are not Numbers
- How do we get this data into numeric format

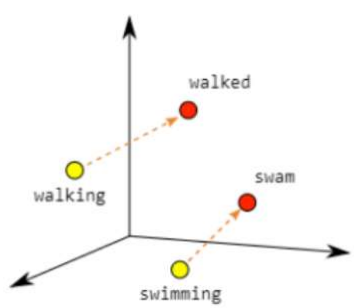
NLP Embeddings

- Embeddings allow us to represent Text with Numbers
- Text is positioned in a high dimensional space
- Similar terms are closer in distance (Euclidean)
- BERT provides us Rich Contextualized Embeddings that understand language

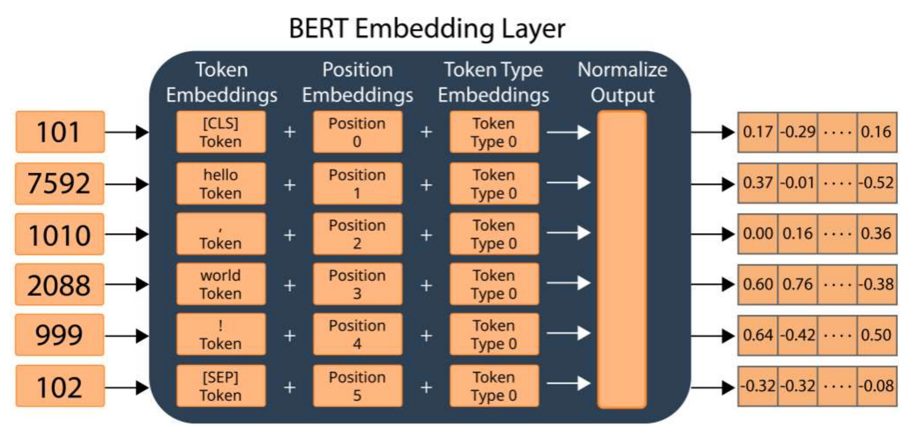
- During Training BERT embeddings are being fine tuned and optimize for the down stream classification problem



Male-Female

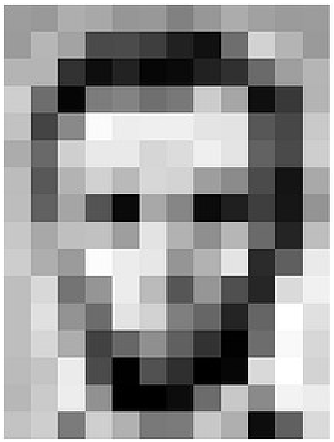


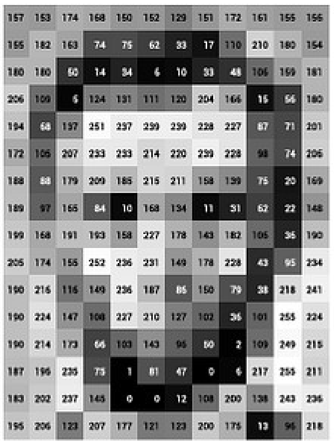
Verb Tense



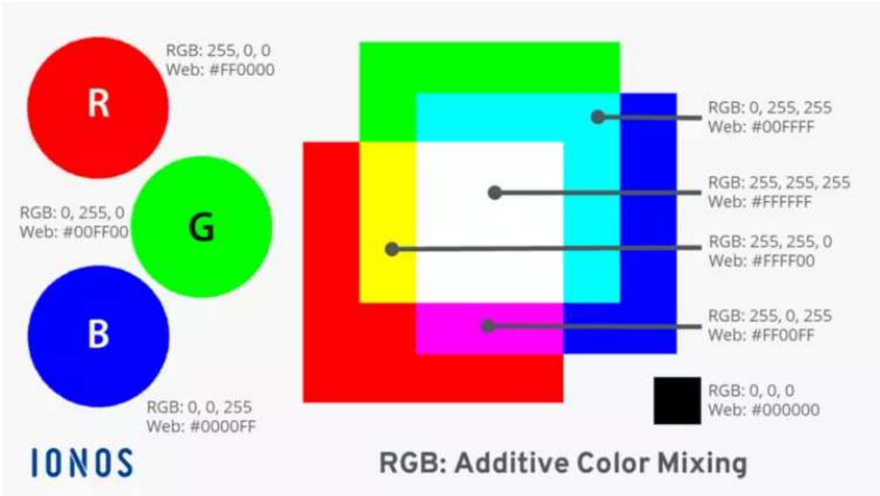
Numeric Representation of images

- Images are represented by an area of Pixels
- Each pixel has a color, each color can be represented by numbers





157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218



Feature Extraction in Computer Vision

- Convolution Filters are applied over images
- Backpropagation optimizes Layer of Convolution Filters for Problem
- Convolution Filter determines feature space
- https://youtu.be/KuXjwB4LzSA?si=R8vKh9LylQ0p268f
- Applying Photo effect Filters

Model Optimization

- Done through Backpropagation
- Minimizing Loss function
- Loss function provides us Gradient to recompute Weights and Biases
- Cross Entropy Loss: Difference between Predicted distribution by model, True distribution of Labels

- Toxic Comments: Binary CrossEntropy

- Animal Classification: Sparse Categorical CrossEntropy

ML Step # 7: Model Evaluation

- Evaluation is done by calculating the percentage of labels accurately predicted by model
- Accuracy
- Precision – Recall
- ROC-AUC
- Confusion Matrix

- Thanks to https://github.com/gnovack (BERT Visual)
- https://www.youtube.com/@venelin_valkov