Embedding

One hot encoding → is something which is like for one feature have multiple categorical data that is like one hot encoding vector.

Encoding : is processing of selecting or same, i.e meaning initial numerical representation of the ~~item or food~~ data to train the model.

Eg: We are developing model can given suggestions based on customer selected food ie

Pizza

1	0	0	0	0	0	0	0	0
Pizza	Biscuit	Item3	Item4	item5	item6	item7	item8	item9

item5

0	0	0	0	1	0	0	0	0
Pizza	Biscuit	Item3	Item4	item5	item6	item7	item8	item9

Drawback

Here there are around 5000 types of food item varieties creating encoding in this way is

1. Large number of vectors means lot of weights for neural network
   eg: M no of entries in encoding and there are N vectors then it has to train on M×N for each layer of neural network
2. Number of data points → weights for model increases accordingly to train effectively
3. Amount of computation also increases
4. Memory for accelerators increases to train weights.
5. Difficulty to support on device Machine Learning (ODML) is hard to implement

need to be focused on making your model smaller & will want to decrease the number of weights.

Embedding

Embedding Space & Static Embedding

Embedding is vector representing data in embedding space

Model find potential embedding by projecting high dimensional ~~data~~ space of initial to lower-dimensional space.

what is embedding space

suppose there are different food dishes & are placed in multidimensional room. each room has similar dishes placed or with the same characteristics are placed together

1. Items as single point in open space
   ↳ Word or food items

2. Dimension This space has multiple dimensions which are like different characteristics/properties

3. Location Matter: The closer two items are tends to have similar properties
   Eg: Shawarma and hot dog are similar as they are considered as sandwiches.
   One might be sweetness & other one would be liquidness.

4. Machine learning role
   Models create these embedding spaces by analyzing data and figuring out how good they can placed in space

B Real word Embedding

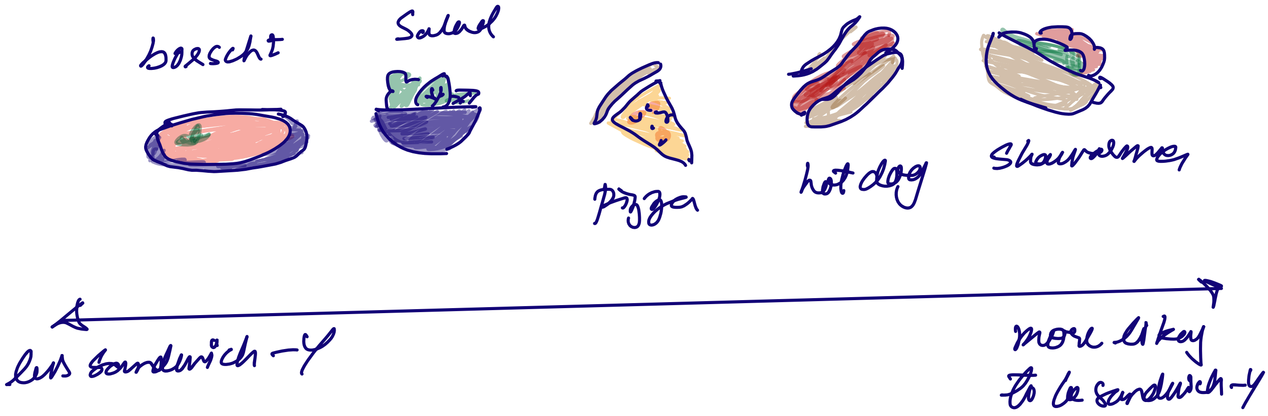
In real world embedding space are d-dimensional, where d is much higher than 3 So less the d-value or dimensions then relationship between data point are less intuitive

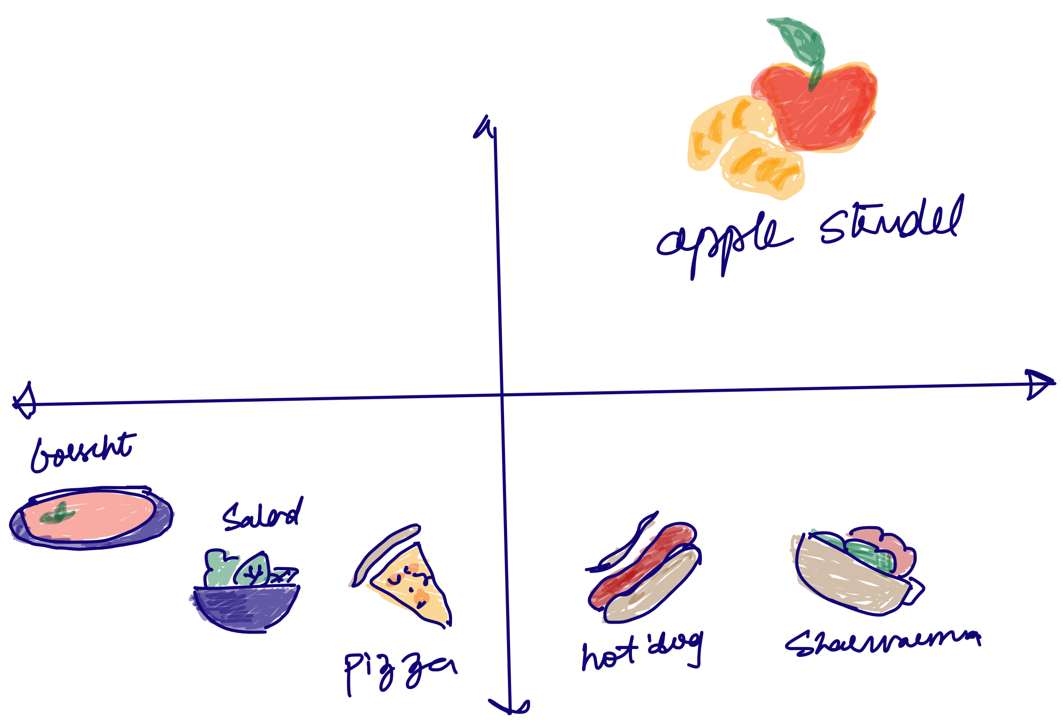
For word embedding majorly the "d" value is 256, 512, 1024 etc

Embedding makes it easier to machine learning on large features

Eg: one-dimensional data

of dishes is " hot dog, pizza, salad, shawarma, and borscht

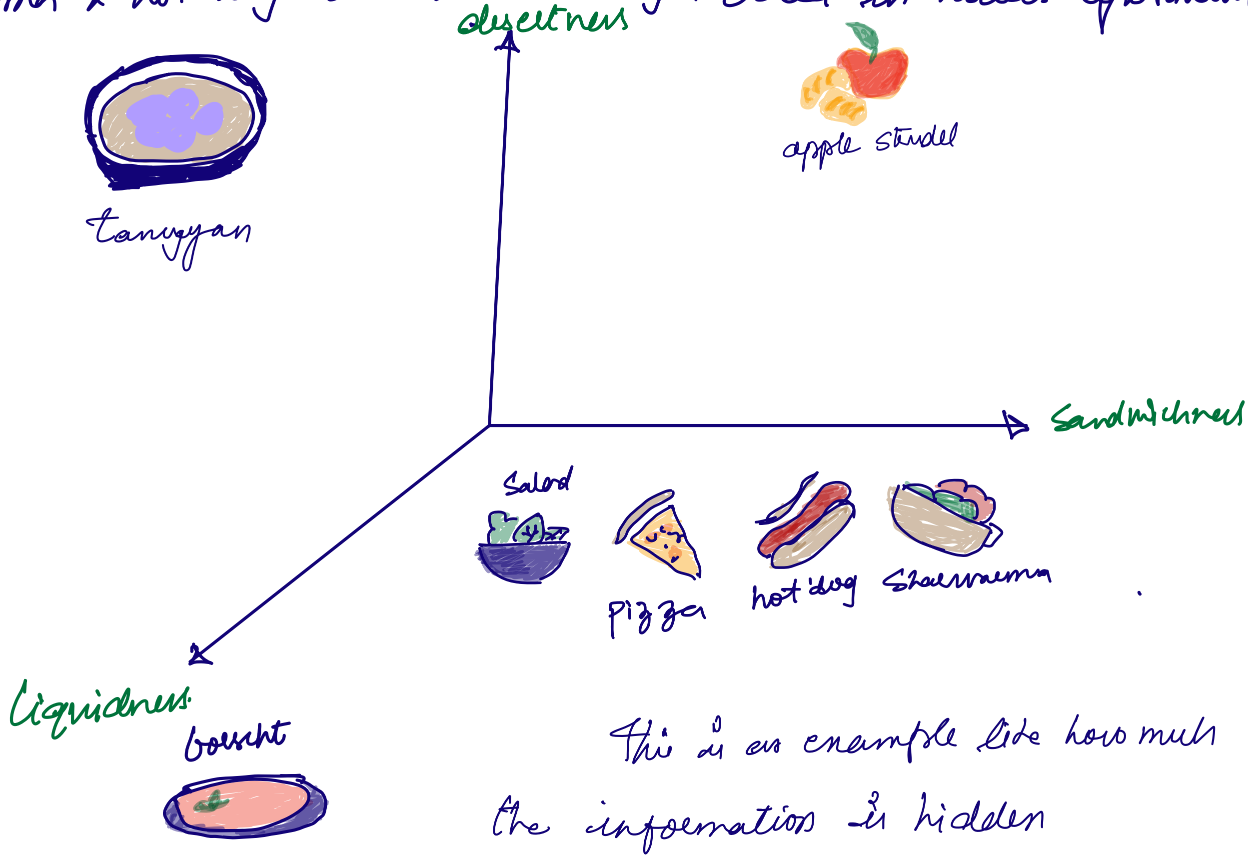




An embedding represent in N-dimensional space with a floating point number typically in range -1 to 1 (or) 0 to 1

An embedding is calculated between ~~two~~ any two items is measure of relative similarity between those two items.

Two things are closer to each other, like Shawarma & hot dog are more closely related in models representation



This is an example like how much the information is hidden

<u>Real World Embedding</u>

In real world the embedding are d-dimensional where d is much higher than 3 and below this are not much intuitive

Usually ML practitioners sets specific task & no of Embedding dimensions, then model tries to arrange the examples to close to space with specified number of dimensions or tunes no of dimensions, if 'd' is not fixed.

Individual dimensions are understandable like liquidness or desertness

Embedding will be specific to task, if task differs they differs

Eg:

Embedding generated ~~from~~ by vegetarian Vs no-vegetarian Classification Model can be different from the embedding generated from by a model which is based on dishes by season, time of day

"Cereal" & "breakfast" will be close when compare to veg Vs nonveg

Static Embedding

While embedding differs from task to task - One specific task about getting content of "word"

Models trained ~~on~~ to predict content of a word assumes words appear in same content are semantically related.

Eg: training data containing below sentences

{ ⇒ They rode burro down to the Grand canyon
⇒ They rode horse down into the canyon }

Here Horse appears in similar content to burro

Eg: Word to vec

↳ This model trains on corpus of document to obtain single global embedding per word.

When each word has single embedding vector then this "$\boxed{\text{STATIC EMBEDDING}}$"

When Static embedding are trained they encode some degree of semantic Information

Embedding are not limited to words, Images, audio, image

OBTAINING EMBEDDING

Dimensionality reduction technique

There are many mathematical Techniques that captures the important structure of high dimensional space in a low-dimensional space

In theory any of these can be used to create embedding for Machine learning.

(1) PCA → Principle component Analysis

↳ It is used to create word embedding

↳ given a set of instances like bag of words vector PCA finds highly correlated dimensions that can be collapsed into single dimension

Training an embedding as part of a neural network

→ In any neural network the parameters will be optimized during training to minimize loss of the model's predictions using a $\boxed{\text{softmax}}$ loss

→ creating a hidden layer of size "d" in your neural network that is said to "Embedding Layer"

Here d → represents hidden layers and Number of nodes

→ and number of dimensions in embedding space

Contextual Embedding

limitation of word2vec is words can mean different in different context

Eg: "Orange" → when we train it will be closer to colour than fruit, when trained on word2vec dataset

Contextual embedding is something developed to overcome this limitation

Contextual embedding allows a word to be represented by multiple words with surrounding words with itself

orange would have different embedding for every unique sentence in dataset

Some methods to create these kind of embedding.

⇒ ELMo, → It takes static embedding of an example such as word2vec, vector ~~in~~ for a word in a sentence and transform it by function which can include the words around it

This generates contextual embedding

These static embedded are aggregated with embedding taken from reading front-to-back, back-to-front

⇒ BERT → Models mask part of the sequence that the model takes as input

⇒ Transformer Model uses a self attention layer to weight the relevance of other word in a sequence to each individual word. also add the relevant column from a positional embedding Matrix to each previously learned token embedding, element by element to produce the input embedding that is feed into the rest of the model for inference

This input embedding, unique to each distinct contextual sequence, is a contextual embedding.

This models are language model, contextual embedding are useful to in other generative tasks, like images.

An embedding of RGB pixel value in a photo of house provides more information to the model when combined with positional Matrix representing each pixel & some encoding of the neighbouring pixels creating contextual embedding, than the original static embedding of the RGB values alone