Sofmax

$$s\left(x_i\right) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

Soft RELU (Softplus)

y = ln(1 + e^x)

$$SVMLoss = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + 1)$$

$$CrossEntropyLoss = -(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i))$$

Huber Loss (MAE becomes MSE near minimum)

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for} |y - f(x)| \le \delta, \\ \delta\, |y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

Likelihood (MLE that observed data D is most probable under model w/params θ to make inference about population that generated sample).

$$\mathcal{L}(\theta|D) = f(D|\theta) = \prod_{i=1}^{N} f(x_i|\theta)$$

log likelihood:

$$l(\theta) = \ln f(D|\theta) = \ln \prod_{i=1}^{N} f(x_i|\theta) = \sum_{i=1}^{N} \ln f(x_i|\theta) \checkmark$$

therefore, $\hat{\theta} = \underset{\theta}{\text{argmax}}\; l(\theta)$

Bayesian estimate (using prior knowledge, compute posterior PDF)

$$\underbrace{p(\theta|D)}_{\text{posterior distribution}} = \frac{\underbrace{p(D|\theta)}_{\text{likelihood function}}\, \underbrace{p(\theta)}_{\text{prior distribution}}}{\int p(D|\theta)\, p(\theta)\, d\theta}$$

Cosine distance

$$\frac{x \bullet y}{\sqrt{x \bullet x}\sqrt{y \bullet y}}$$

Naive Bayes

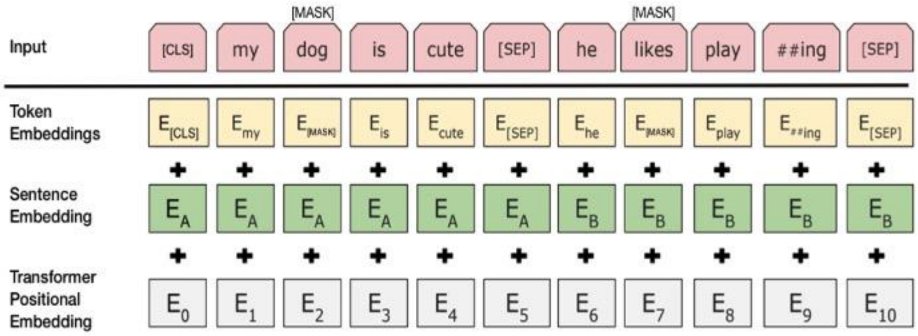
$$P(L \mid \text{features}) = \frac{P(\text{features} \mid L) P(L)}{P(\text{features})}$$

$$T - statistic = \frac{Observed\ value - hypothesized\ value}{Standard\ Error}$$

Euclidean (Minkowski 2)

$$\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

$$Stamdard\ Error = \sqrt{\frac{2 * Variance(sample)}{N}}$$



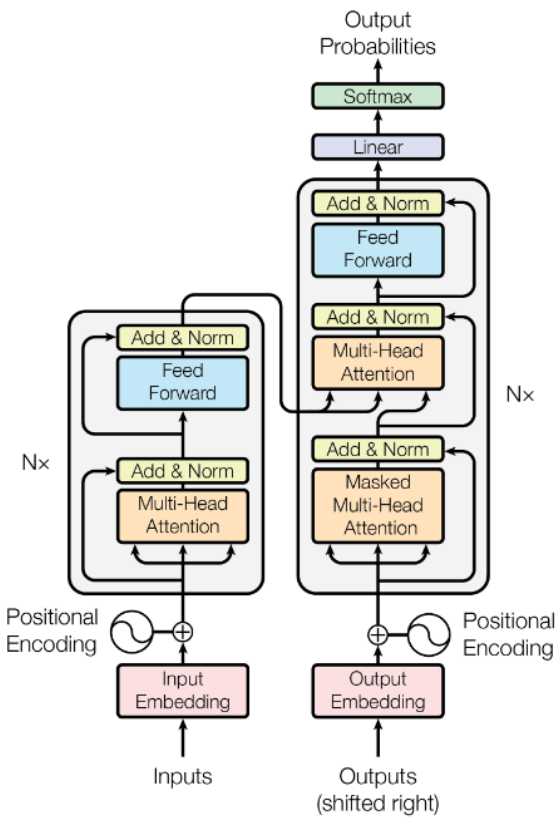
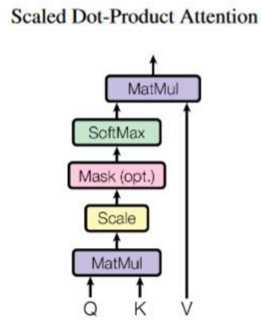
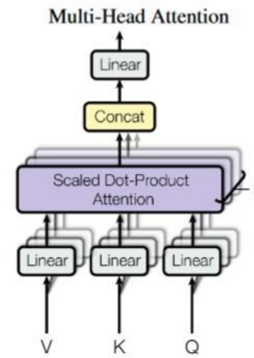


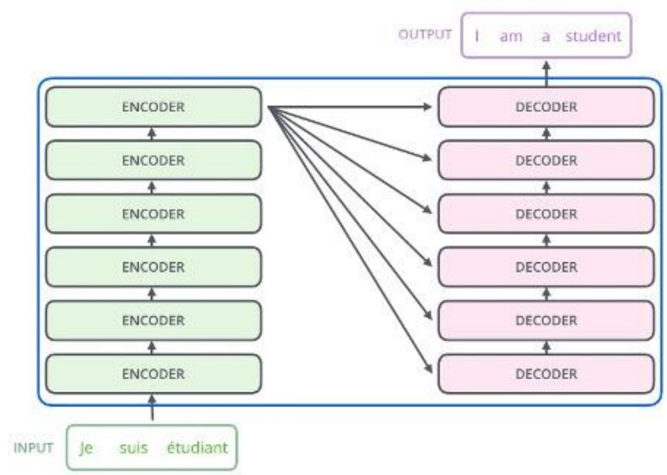
Figure 1: The Transformer - model architecture.

Self-Attention

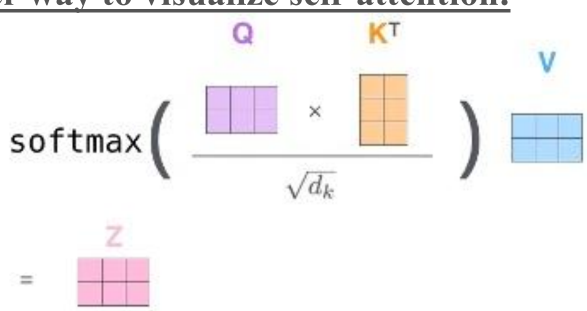




Encoder-Decoder Attention



Another way to visualize self-attention:



Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision = TP / (TP + FP) – how often clf correct when predicting positive

Recall = TP / (TP + FN) – how often clf correct for all positive instances (aka **sensitivity**, aka **TPR** (True Positive Rate)

$$F_\beta = (1+\beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}.$$

β is the weight of **recall** (gives more importance to precision if β < 1)

Specificity, **TNR** (true negative rate) = TN / (TN + FP)

FPR(False Positive Rate) = 1 - Specificity = FP / (TN + FP)

Prediction bias = "average prediction" – "average observations". Ideally should be 0, and a significant nonzero prediction bias => bug in the model because the model is wrong about how frequently positive labels occur

AUC (ROC curve): Plots sensitivity(TPR) and (1-specificity)(FPR)

PR-AUC: precision-recall curve

Ranking (recommend) – **prec@k**, **rec@k**, mean reciprocal rank **MRR** (1/position of first relevant item – *interpretable*!), mean ave. precision **mAP** (averages prec.@k at each relevant item position, rewards model that *puts more relevant stuff at top*, ideal=1, some interpretability), normalized discounted cumulative gain **nDCG** (cumulative gain = *sum of relevant items*, *discounted* for each position, *normalized* by position 1, *not interpretable* but takes into account *numeric relevancy*, tricky on practice => *not used often*)

Text generation – **BLEU** (similarity of machine-transl. text to high quality translation = *precision-based n-gram overlap* w/brevity penalty), **ROUGE-N** (*recall-based n-gram overlap* in machine transl. or summary), **METEOR** (harmonic mean of *unigram precision and recall* *w/stemming and synonymy matching*, along w/exact word matching, fixes issues w/BLEU), **CIDEr** (Consensus-based Image Description Evaluation - evaluates for image Li *how well a candidate sentence Ci matches a set of image descriptions*), **SPICE** (Semantic Propositional *Image Caption Evaluation*)

Image generation – **FID** (Frechet Inception Distance score - *distance between feature vectors* of real and generated images), **inception score** (takes a l*ist of GAN-generated images*, returns *one score 0-inf* re how good they are)

Ensemble learning:

a) **boosting** to reduce bias (ensemble of **"weak" classifiers** trained consecutively where misclassified data points are given higher weight at next steps, but the overall prob distribution is still 1),

b) **bagging** or bootstrap aggregation to reduce variance (combination of **"strong" learners** with same vote trained on partial data each using resampling with replacement).

Online Evaluation

- Based on **business objectives**
- **CTR** - # clicks / # times shown
- **Recommender** - # recommend. made / # recommend. accepted
- **Chatbot** – closure rate, # human interventions, # turns per query
- **Video recommendations** - watch time, # videos watched
- **Harmful content** - # harmful posts not prevented / all posts, # posts appealed and reversed / # harmful posts detected, # detected vs. reported posts (proactive rate)

A/B Testing

Simplest form of **randomized controlled experiment**
Two **samples A and B** are compared – they are **similar except for one variation**.
E.g. two versions of web page / product – **which leaves max impact** on business metrics

1. **Hypothesis Testing**

a) Null hypothesis, H_0: no difference between the control and test groups.

b) Alternative Hypothesis, H_a: concept / <u>educated guess to be </u>**verified.**

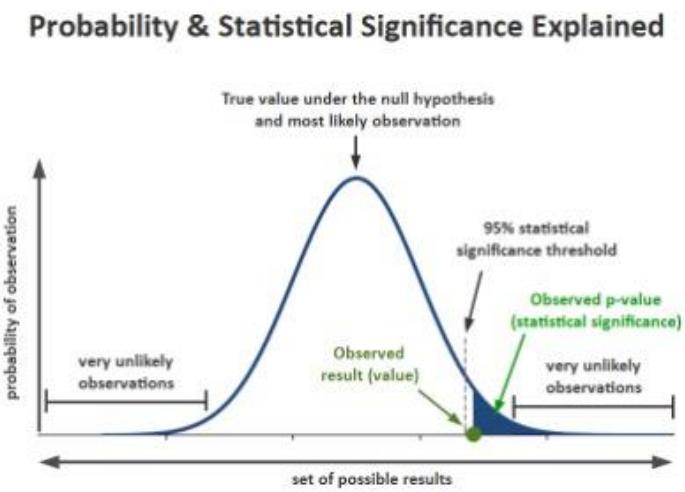
2. Create **Control Group** (no change) and **Test Group** (modified product)
Avoid **selection bias** (<u>random sampling)</u>, **under-coverage bias (**sample size).

3. **Conduct A/B Test, collect data**

4, **Statistical significance**

   1.**Type I error**: rejecting H_0 when it is true ("You are pregnant" to a man)
   2.**Type II error**: not rejecting H_0 when it is false ("You are not pregnant" to pregn. woman)

To avoid these errors – determine statistical significance of our test.



1. **Significance level (α) =** 0.05, probability of Type I error
2. **P-Value** = smallest significance at which H_0 is rejected. Smaller p-value - stronger evidence for H_a.
   If p-value < significance level 0.05, reject H_0.
3. **Confidence interval =** u r 95% confident that the result is accurate

Next, calculate t statistics:

$$T - statistic = \frac{Observed\ value - hypothesized\ value}{Standard\ Error}$$

$$Stamdard\ Error = \sqrt{\frac{2 * Variance(sample)}{N}}$$

In Python – **scipy.stats.ttest_ind**

Avoid: **Invalid hypothesis, Testing too Many Elements Together, Ignoring Statistical Significance** (doesn't matter what you feel), **not considering the external factor** (new website on days highest traffic)
A/B testing works best when **testing incremental changes** (UX changes, new features, ranking, and page load times - compare pre and post-modification results), not major changes