*What is perplexity? What is its place in NLP?*

Perplexity is a way to express a degree of confusion a model has in predicting. More entropy = more confusion. Perplexity is used to evaluate language models in NLP. A good language model assigns a higher probability to the right prediction.

*What is the problem with ReLu?*

* Exploding gradient(Solved by gradient clipping)
* Dying ReLu — No learning if the activation is 0 (Solved by parametric relu)
* Mean and variance of activations is not 0 and 1.(Partially solved by subtracting around 0.5 from activation. Better explained in fastai videos)

*What is the difference between learning latent features using SVD and getting embedding vectors using deep network?*

SVD uses linear combination of inputs while a neural network uses non-linear combination.

*What is the information in hidden and cell state of LSTM?*

Hidden stores all the information till that time step and cell state stores particular information that might be needed in the future time step.

*Number of parameters in an LSTM model with bias*

4(𝑚*h+h²+h*) where 𝑚 is input vectors size and *h* is output vectors size a.k.a. hidden

The point to see here is that *mh* dictates the model size as m>>h. Hence it's important to have a small vocab.

*Time complexity of LSTM*

seq\_length\*hidden²

*Time complexity of transformer*

seq\_length²\*hidden

When hidden size is more than the seq\_length(which is normally the case), transfomer is faster than LSTM.

*What are the limitation of Adam optimiser?*

“While training with Adam helps in getting fast convergence, the resulting model will often [have worse generalization performance than when training with SGD with momentum](https://arxiv.org/abs/1705.08292). Another issue is that even though Adam has adaptive learning rates its performance improves when using a good learning rate schedule. Especially early in the training, it is beneficial to use a lower learning rate to avoid divergence. This is because in the beginning, the model weights are random, and thus the resulting gradients are not very reliable. A learning rate that is too large might result in the model taking too large steps and not settling in on any decent weights. When the model overcomes these initial stability issues the learning rate can be increased to speed up convergence. This process is called learning rate warm-up, and one version of it is described in the paper [Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour](https://arxiv.org/abs/1706.02677).” — from [iprally](https://www.iprally.com/news/recent-improvements-to-the-adam-optimizer" \t "_blank)

*How is AdamW different from Adam?*

AdamW is Adam with L2 regularisation on weight as models with smaller weights generalise better

*Should we do cross-validation in deep learning?*

No.

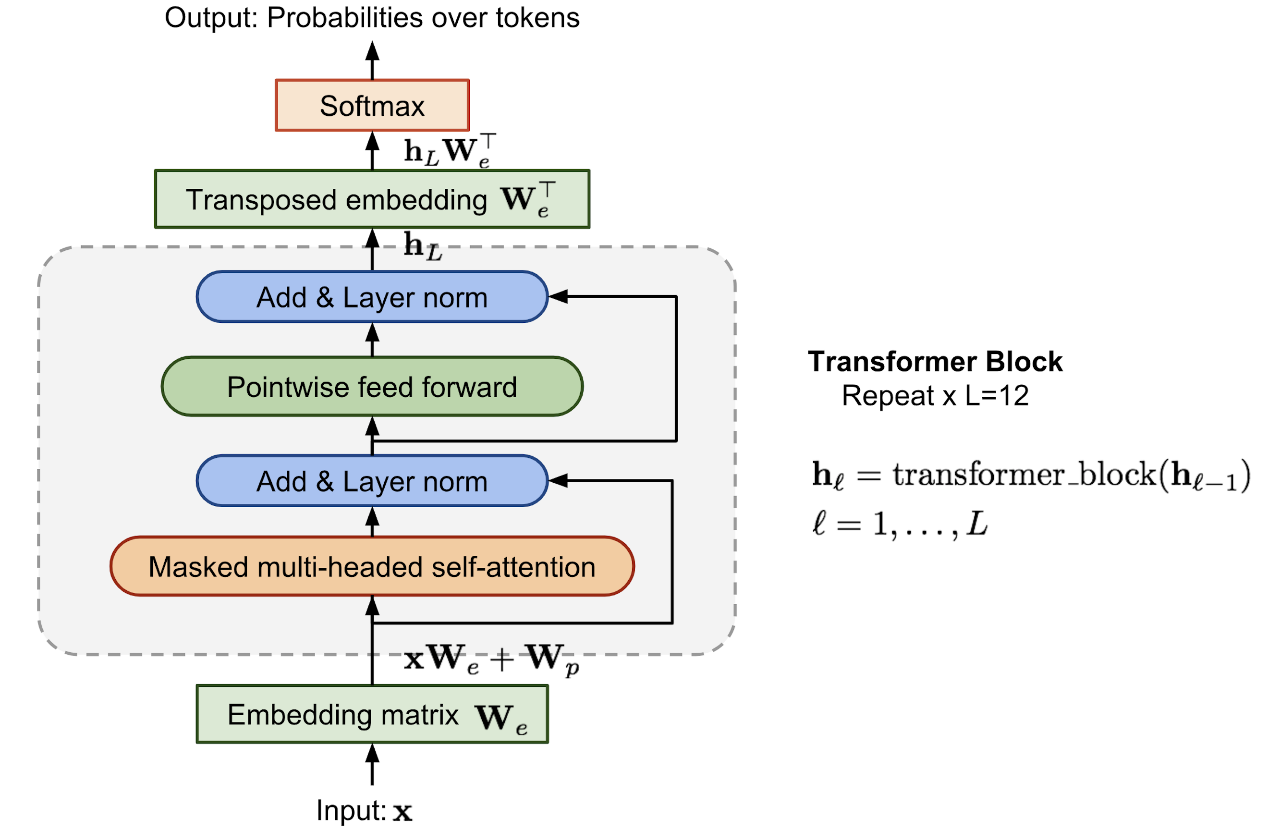
The variance of cross-folds decrease as the samples size grows. Since we do deep learning only if we have samples in thousands, there is not much point in cross validation.

*Difference between BatchNorm and LayerNorm?*

BatchNorm — Compute the mean and var at each layer for every minibatch

LayerNorm — Compute the mean and var for every single sample for each layer independently

*Why does the transformer block have LayerNorm instead of BatchNorm?*



Looking at the advantages of LayerNorm, it is robust to batch size and works better as it works at the sample level and not batch level.

*What are the tricks used in ULMFiT? (Not a great questions but checks the awareness)*

* LM tuning with task text
* Weight dropout
* Discriminative learning rates for layers
* Gradual unfreezing of layers
* Slanted triangular learning rate schedule

This can be followed up with a question on explaining how they help.

*Why transformer perform better than LSTM?*

<https://medium.com/saarthi-ai/transformers-attention-based-seq2seq-machine-translation-a28940aaa4fe>

*Funny questions: Which is the most used layer in transformer?*

Dropout

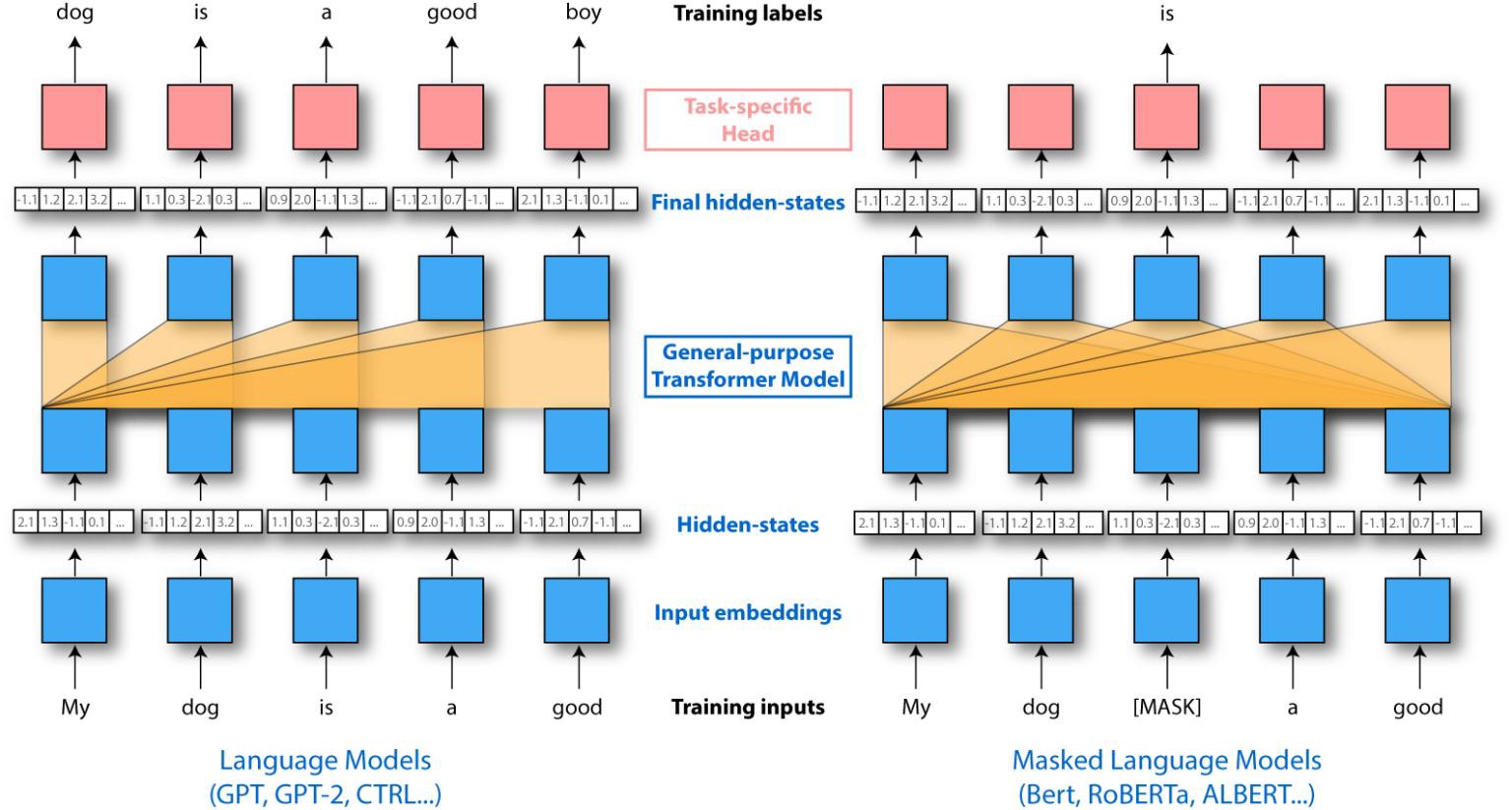
*Trick question: Tell me a language model which doesn’t use dropout*

ALBERT v2 — This throws a light on the fact that a lot of assumptions we take for granted are not necessarily true. The regularisation effect of parameter sharing in ALBERT is so strong that dropouts are not needed. (ALBERT v1 had dropouts.)

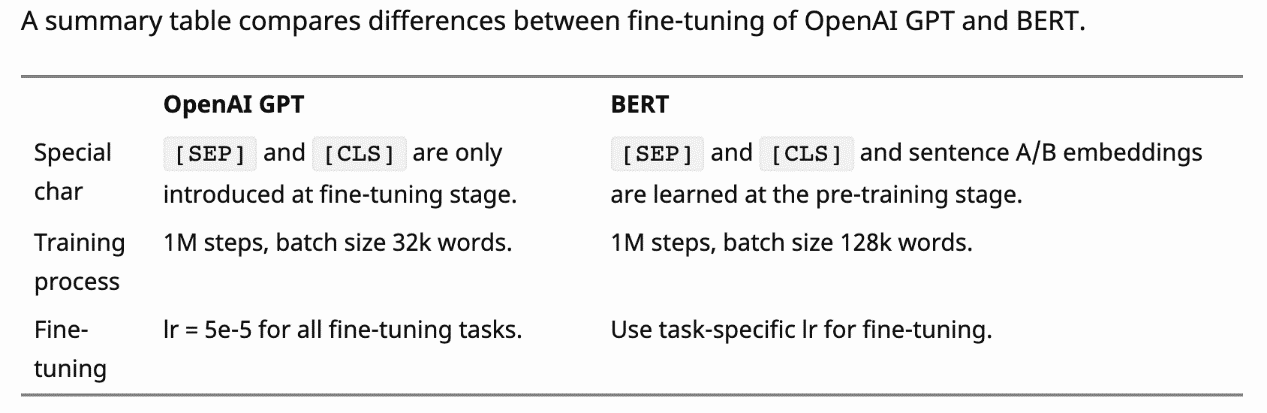
*What are the differences between GPT and GPT-2? (From Lilian Weng)*

* [Layer normalization](https://arxiv.org/abs/1607.06450) was moved to the input of each sub-block, similar to a residual unit of type [“building block”](https://arxiv.org/abs/1603.05027) (differently from the original type [“bottleneck”](https://arxiv.org/abs/1512.03385), it has batch normalization applied before weight layers).
* An additional layer normalization was added after the final self-attention block.
* A modified initialization was constructed as a function of the model depth.
* The weights of residual layers were initially scaled by a factor of 1/√n where n is the number of residual layers.
* Use larger vocabulary size and context size.

*What are the differences between GPT and BERT?*



* GPT is not bidirectional and has no concept of masking
* BERT adds next sentence prediction task in training and so it also has a segment embedding



*What are the differences between BERT and ALBERT v2?*

* Embedding matrix factorisation(helps in reducing no. of parameters)
* No dropout
* Parameter sharing(helps in reducing no. of parameters and regularisation)

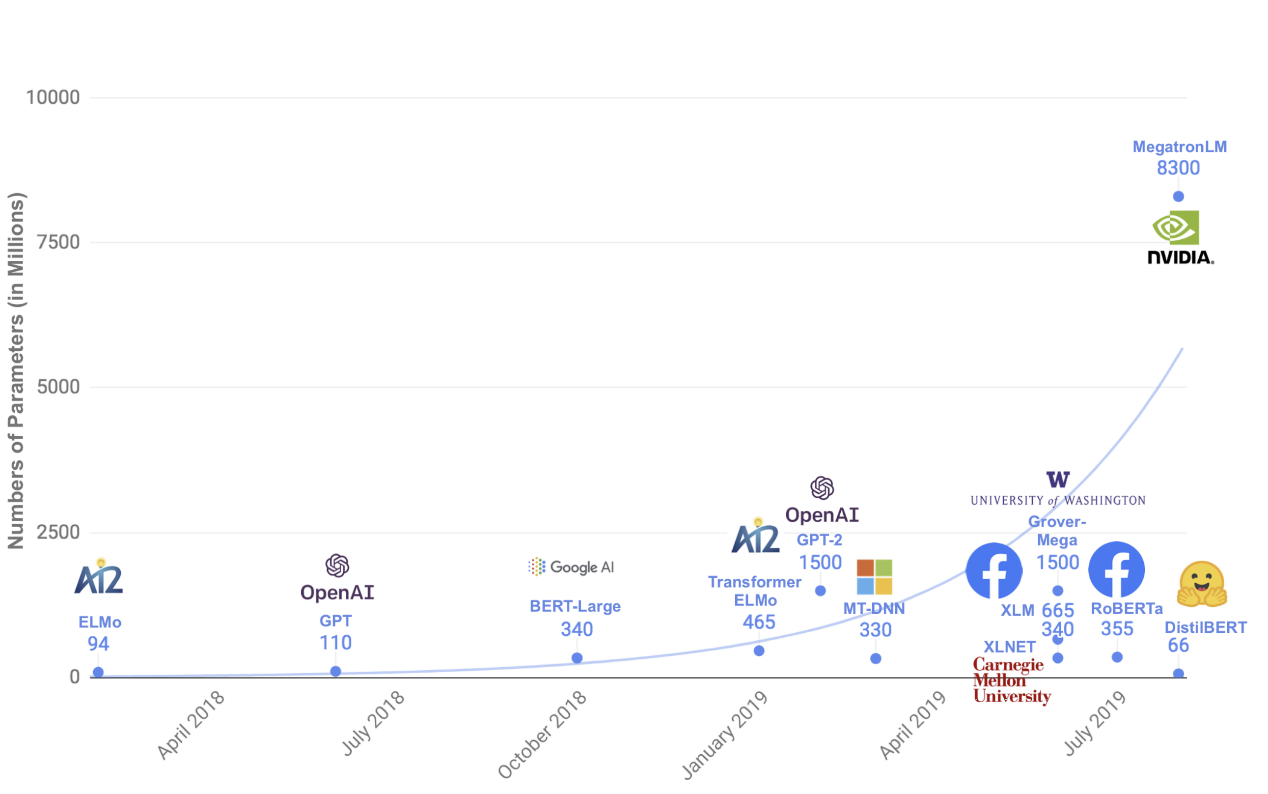
*How does parameter sharing in ALBERT affect the training and inference time?*

No effect. Parameter sharing just decreases the number of parameters.

*How would you reduce the inference time of a trained NN model?*

* Serve on GPU/TPU/FPGA
* 16 bit quantisation and served on GPU with fp16 support
* Pruning to reduce parameters
* Knowledge distillation (To a smaller transformer model or simple neural network)
* Hierarchical softmax/Adaptive softmax
* You can also cache results as explained here.

*Given this chart, will you go with a transformer model or LSTM language model?*



*How would you make a sentiment classifier?*

This is a trick question. The interviewee can say all things such as using transfer learning and latest models but they need to talk about having a neutral class too otherwise you can have really good accuracy/f1 and still, the model will classify everything into positive or negative.

The truth is that a lot of news is neutral and so the training needs to have this class. The interviewee should also talk about how will he create a dataset and his training strategies like the selection of language model, language model fine-tuning and using various datasets for multi-task learning.