a Consider the following statements regarding Optimization and Regularization of neural networks. Mark each of the statements which are true.

4.0 points · Multiple choice · 4 alternatives

In Layer Normalization, the mean and standard deviation statistics are independent of the batch dimension.	2.0 points
During training, Dropout replaces the value of the 'dropped' neurons with new, random samples from a unit Gaussian.	-2.0 points
The Adam optimizer uses both momentum and an adaptive learning rate.	2.0 points
Applying an L1 regularization on the weight parameters is identical to multiplying the parameters by a factor close to 1 at each step (weight decay).	-2.0 points

b Consider the following statements regarding Convolutional Neural Networks (CNNs). Mark each of the statements which are true.

4.0 points · Multiple choice · 4 alternatives

If you apply a convolutional layer with kernel size 3 on an image of size 12x12 (no padding, stride 1), the output of this operation will be the size 9x9.	-2.0 points
Standard CNNs like the original AlexNet architecture are rotation invariant.	-2.0 points
Applying 2 3x3 convolutions have fewer parameters than one 5x5 convolution, when the input and output channel size is the same.	2.0 points
An inception module applies a 1x1, 3x3, 5x5 and a max pooling operation on the same input feature map.	2.0 points

c Consider the following statements regarding Transformers and Multi-Head attention. Mark each of the statements which is true.

4.0 points · Multiple choice · 4 alternatives

Self-attention has a linear complexity over the sequence length.	-1.67 points
Transformers for language tasks such as machine translation are permutation equivariant.	-1.66 points
The BERT model is pretrained by autoregressive language modeling.	-1.66 points
In multi-head attention as applied in Transformers, each head uses a different set of key, query and value vectors.	4.0 points

a If the input to this ResNet block (x_t) is distributed according to a Gaussian with zero mean and a variance of σ_t^2, what distribution does x_{t+1} follow?

Hint: you can assume that the input and output variables of a convolution are statistically independent. However, the input and output distributions might have dependencies with their mean and standard deviation.

5.0 points · Open · 1/4 Page

Model answer

With the kaiming initialization in the ResNet block, we know that if the input variance is σ_t^2, the output variance of the last linear/convolution layer is as well σ_t^2. A common mistake was assuming that the output has the weight variance, $2/d_x$, but we need the feature variance here.

Taking the assumption stated in the question that the output of the ResNet block being statistically independent of its input (which at init commonly holds in practice), we can use the variance formula given in the cheat sheet and used several times in deriving the initialization techniques: Var(X+Y)=Var(X)+Var(Y). This gives us the variance of $Var(x_t) + Var(F(x_t)) = \sigma_t^2 + \sigma_t^2 = 2\sigma_t^2$, or an standard deviation of $\sqrt{2}\sigma_t$.

+5 points

Gaussian distribution with mean 0 variance $2\sigma_t^2$

+1.5 points

Gaussian with mean 0

+3.5 points

Gaussian with variance $2\sigma_t^2$

b What problem could this cause in a deep architecture? How does the ResNet architecture overcome it?

6.0 points · Open · 1/2 Page

Model answer

The problem of this block is that the feature variance explodes when going deeper in the architecture. For instance, if we start with a variance of 1, the output of the first ResNet block has 2, the one after that 4, then 8, etc. The variance grows exponentially, which makes learning impossible for deep ResNets.
The solution was hinted by the block which was missing from the standard ResNet above: Batch Norm. The normalization step scales down the variance before any linear layer/non-linear activation, making learning possible again. Note that this is not specific to Batch Norm, but also LayerNorm or other normalization techniques would have the same effect.

This question could also be answered correctly when the answer in a) was incorrect. Since we add a statistically independent variable to another one, the variance can only increase over layers.

+3 points

Problem: exploding variance with increasing depth.

+3 points

Solution: BatchNorms before any ResNet block and final classification layer

+2 points

For the first question of 2b, the correct answer is exploding variance with increasing depth. However, for those students who give answers like Unstable/consistent variance/gradient exploding with increasing depth, which are not completely right but almost close to the right answer, we also give 2 points as rewards.

a We train both networks on the CIFAR10 dataset using an SGD optimizer. We observe that, despite network 1 having much more parameters, network 2 outperforms network 1. On the test set, network 1 achieves around 10% accuracy, while network 2 achieves 38%. Please elaborate on why network 2 outperforms network 1 in this setup, and what optimization issues network 1 has.

6.0 points · Open · 1/2 Page

+6 points
gradient vanishing

+3 points
Kaiming initialization is not sufficient for Network 1 with sigmoid activation.

0 points
The loss of non-linearity is incorrect since network 2 actually has no non-linearity whatsoever.

0 points
Over-fitting is incorrect.
The model doesn't learn anything, even the training stays at random 10%.

0 points
common mistakes

b Propose one solution for how you could achieve a higher accuracy with network 1 than network 2, while keeping 5 linear layers and Sigmoid as an activation function. Explain how this method overcomes the previous issues of network 1.

5.0 points · Open · 2/5 Page

+5 points

Optimizer with adaptive learning rate (e.g. Adam). Scales the gradient distribution per layer appropriately and allows reasonable step size per layer.

+5 points

Batch Normalization. Ensures that the feature variance is propagated throughout the network, and removes any bias that comes from the Sigmoid.

+5 points

Choosing a proper initialization for the network with the sigmoid, e.g., a Xavier uniform initialization with much higher variance.

a As seen above, $\mathcal{L}_{ELBO}$ consists out of two terms. What are these two terms typically called?

3.0 points · Open · 1/2 Page

+1.5 points

Right term: Regularization loss

+1.5 points

Left term: Reconstruction loss

b Briefly explain why the names of the two terms (max 5 sentences per term) provided in part a are appropriate.

4.0 points · Open · 1/2 Page

+2 points

- In the reconstruction loss (left term) our data $\boldsymbol{x}$ is first encoded into $\boldsymbol{z}$ and then decoded to a distribution over the input variable $\boldsymbol{x}$, the log likelihood then tells us how well this distribution matches $\boldsymbol{x}$ and thus how well samples from this distribution would be similar to the input. In other words, how well is the input sample $\boldsymbol{x}_n$ reconstructed.

+2 points

- In the regularization loss we minimize the difference between the distribution $q(z)$ and a prior $p(z)$, such that - on average - the distributions over encodings match the prior. Thus, we restrict the space of the latent variables. As these are just per-sample parameters, and restricting the values a parameter can take is called regularization, we can think of this term as a regularization term.

+1 point

Part of the description of Reconstruction or Regularization is correct.

c Given that we choose the following prior: $p(\boldsymbol{z}) = \mathcal{N}(0, I_D)$
Briefly explain (max. 10 sentences) what kind of effect a high value of β would have on the trained model?

5.0 points · Open · 1/2 Page

+5 points
(all correct) A high value of β places considerable emphasis on the KL term in the ELBO. This applies a stronger constraint on the encoding model to be similar to the prior, which in this case forces the latent representation to be conditionally independent or disentangled.

+1 point
Explained that a high value of β places considerable emphasis on the KL term in the ELBO

+4 points
Explained how the emphasis on the KL term applies a stronger constraint on the encoding model to be similar to the prior, which in this case forces the latent representation to be conditionally independent or disentangled.

+2 points
Partially correct but some parts are missing

a How does the above expression for y_i change for a general stride $S \geqslant 1$? (You may assume that the number of features in the input M is divisible by the stride S.)

4.0 points · Open · 1/4 Page

+2 points

$x_{i+k} \rightarrow x_{iS+k}$

a Write down what it means for the function f to be permutation equivariant, and what is the condition on the function g that ensures permutation equivariance of f. Write one example of such g that can be used in the equation.

4.0 points · Open · 1/2 Page

+1 point

Partially correct equation of permutation equivariance or not writing any equation on permutation equivariance. It's not correct to say that the order does not matter: the order do matter, but when the order of the input features is changed, the the output changes in a predictable (in our case the *same)* way. In symbols: $f(PX, PAP^T) = Pf(X, A)$.

+2 points

Permutation equivariance $f(PX, PAP^T) = Pf(X, A)$

+1 point

Writing the correct condition on g: g must be permutation invariant.

+1 point

Correct examples of g:

$g(x_i, \chi_i) = \frac{1}{|\chi_i|} \sum_{x \in \chi_i} x$

$g(x_i, \chi_i) = sup_{x \in \chi_i}(x)$

b Suppose you are performing classification with a GNN. Write an example of a layer that you can apply at the end of a GNN that reduces the input X to a single feature vector, before eventual linear layers for classification. Describe how this layer ensures the GNN's predictions are permutation invariant.

4.0 points · Open · 1/2 Page

+2 points

An example of a permutation-invariant layer.

+2 points

The explanation for why the proposed layer is permutation-invariant.