Robust Deep learning

Perann Nedjar

December 2025

Week 1

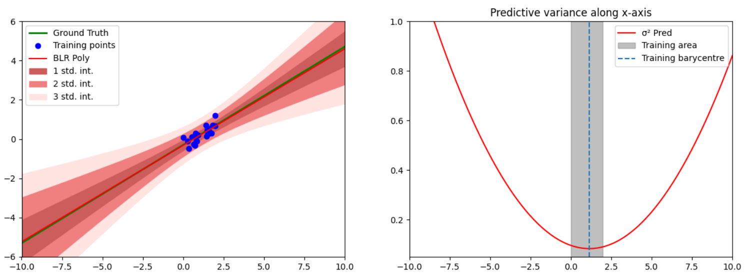

Figure 1: BLR prediction and uncertainity

The formula of predictive variance σ_p is :

$$\sigma_p = \frac{1}{\beta} + \phi(x_*)^T \Sigma \phi(x_*)$$

In a more basic case with $\beta = 1$ and $\alpha = 0$, one has that $\sigma_p = 1 + \phi(x_*)^T \Sigma \phi(x_*)$ and $\Sigma = (\Phi^T \Phi)^{-1}$.

$$\begin{pmatrix} N & \Sigma_{i=1}^N x_i \\ \Sigma_{i=1}^N x_i & \Sigma_{i=1}^N x_i^2 \end{pmatrix}$$

because

$$\phi : x \to \begin{pmatrix} 1 \\ x \end{pmatrix}$$

So $\Sigma = \frac{1}{N \Sigma_{i=1}^N x_i^2 - (\Sigma_{i=1}^N x_i)^2} \begin{pmatrix} N & -\Sigma_{i=1}^N x_i \\ -\Sigma_{i=1}^N x_i & \Sigma_{i=1}^N x_i^2 \end{pmatrix}$

Non-singularity is ensured by Cauchy-Schwartz inequality and iid hypothesis on X_i.

So the predictive variance has a expression given by :

$$\sigma_p^2 = 1 + \begin{pmatrix} 1 & x_* \end{pmatrix} \frac{1}{N\Sigma_{i=1}^N x_i^2 - (\Sigma_{i=1}^N x_i)^2} \begin{pmatrix} N & -\Sigma_{i=1}^N x_i \\ -\Sigma_{i=1}^N x_i & \Sigma_{i=1}^N x_i^2 \end{pmatrix} \begin{pmatrix} 1 \\ x_* \end{pmatrix}$$

$$\sigma_p^2 = 1 + \frac{1}{N\Sigma_{i=1}^N x_i^2 - (\Sigma_{i=1}^N x_i)^2} (\Sigma_{i=1}^N x_i^2 - 2x\Sigma_{i=1}^N x_i + N x_*^2)$$

Which is a second order parabola that reaches its minimum for $x_* = \frac{1}{N}\Sigma_{i=1}^N x_i$

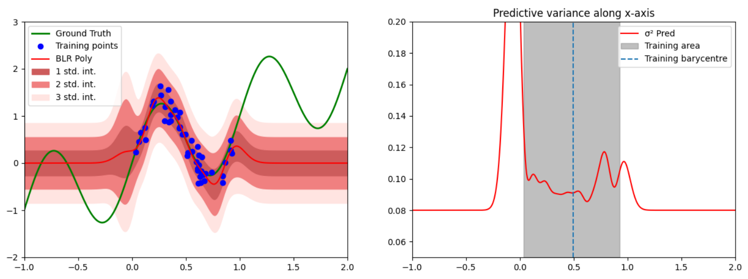

Figure 2: RBF BLR prediction and uncertainity

If the kernel is centered, the term $\phi(x_*)\Sigma\phi(x_*)$ happens to be also centered. It's easy to vizualize in the real case with a standard gaussian kernel, where $\sigma_p^2 = 1 + \frac{e^{-x_*^2}}{\Sigma_{i=1}^N e^{x_i^2}}$, which is centered around 0. This is true even with Σ a matrix, as Σ is psd, it can be vizualize as serie of positive scaling along an orthonormal basis, so it does not entains translation, so it keeps data centered.

To show this, let's write $\Sigma = P\Delta P^T = P\Delta^{1/2}\Delta^{1/2}P^T = P\Delta^{1/2}(P\Delta^{1/2})^T$. By writing the coeffcient of $P = \begin{pmatrix} u_{0,0} & u_{0,2} & \dots & u_{0,M} \\ u_{1,0} & u_{1,2} & \dots & u_{1,M} \\ . & . & \dots & . \\ u_{M,0} & u_{M,2} & \dots & u_{M,M} \end{pmatrix}$

So finaly $\sigma_p^2 = \Sigma_{i=0}^M \Sigma_{j=0}^M \lambda_i u_{j,i} \phi_i(x_*)$. So σ_p^2 converges toward 0 when x* far away from centers of kernels.

Week 2

Laplace approximation result

Now the result is more coherent, the confidence does decrease far from the training data. The predictive distribution is averaged around the MAP. This kind of approximation is unable to capture certain characteristics of the predictive distribution, such as skew for instance.

Variational inference

class VariationalLogisticRegression:

The sampling function generates samples from a Gaussian distribution using the reparameterization trick. The kl divergence function computes the closed-form KL divergence between the variational posterior of the weights and the Gaussian prior as a regularization term.Also, the forward method samples weights and biases and performs a linear transformation on the input using these sampled parameters

class LinearVariational:

Bayesian logistic regression model built on top of the previously defined LinearVariational. The forward method passes the input x through the variational layer and applies a sigmoid activation, producing a probability output for binary classification.

The kl divergence method calls the KL computation of the underlying LinearVariational layer, returning the sum of KL divergences for all weights.

Prediction results : Regarding the prediction result, it is more realistic than MAP, as uncertainity decreases far from training distribution. Compared to laplace approximation, VI provides a full posterior approximation whereas laplace's just do a gausssian approximation around the MAP.

MC dropout

With the MC dropout method, data is well classified (100% accuracy). The model is able to correctly catch the non linearity of the dataset.

Uncertainty is well modeled near the training data, but the model gets overconfident far from it. This is partly because ReLU networks extrapolate linearly outside the training region, so uncertainty estimates may not reflect true confidence in unseen regions

MC dropout has several advantages over bayesian logistic regression with Variational inference. First it is easy to implement and can be applied to any neural network, so it allows to catch more complex data organization (non linear separable dataset for instance). This simplicity also makes it more computationally efficient. On the other hand, on smaller and linear separable dataset, bayesian logistic regression with variational inference can provide better and more interpretable results.

Week 3

We train a classic ResNet-18 architecture on the CIFAR-100 dataset, containing images classified in 100 different category. After training with Adam optimizer on 100 epochs, we reach a test accuracy of 72.55%.

We obtain the following learning curves :

The model logically achieves better performance on the training data than the test data. It does not seem to overfit too much as the performance on the test set do not decreases. Nevertheless, performance actually doesn't improve

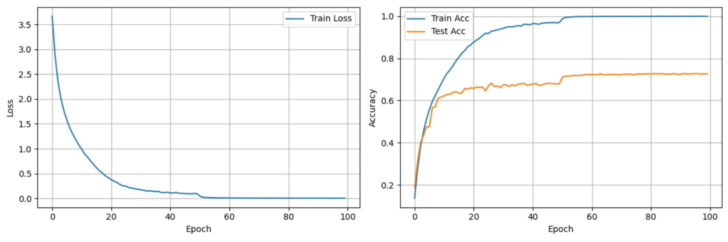

Figure 3: Learning curves

after 50 epochs, the model could have been trained on less epochs for same performances.

We will use the CIFAR-10 for near OOD and SVHN dataset for far OOD. We compute several OOD metrics and plot their ROC AUC curve on the following plot:

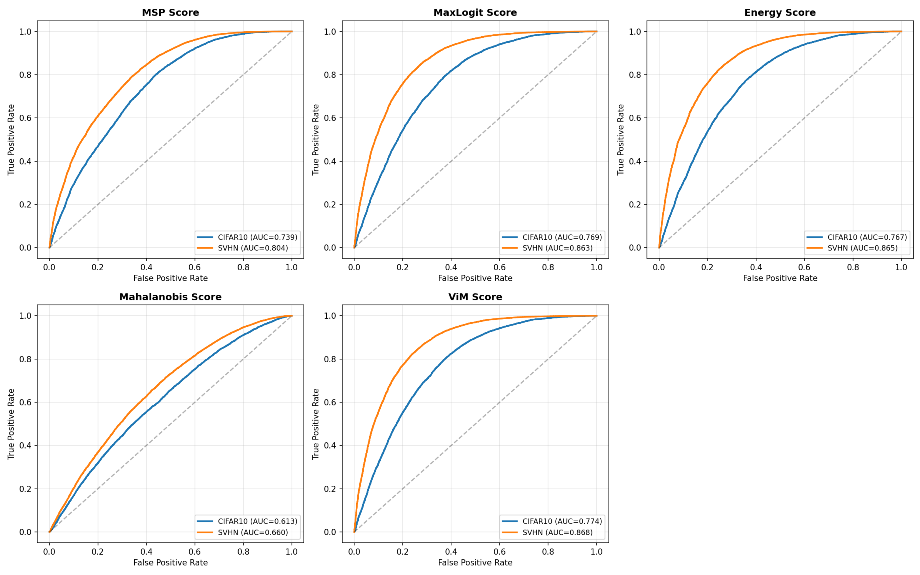

Figure 4: OOD Metrics

As expected, classification is better for images of the far OOD dataset (SVHN).

The worst metric for separation here is the Mahalanobis score, but it stil performs above random classifier. The metric with the best ROC AUC Score is ViM, max logit and Ernergy score have slightly lower performances.

We now focus on Neural collapse phenomenon, by plotting the within class variance for each class and the cosine similarity between class means :

Then we analyse classifier class mean alignement ;

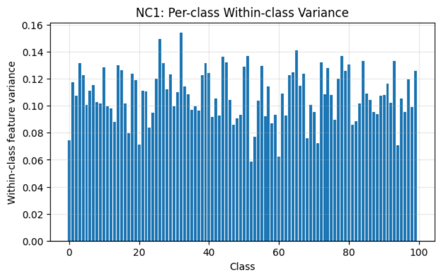

Figure 5: Within-class variance (NC1)

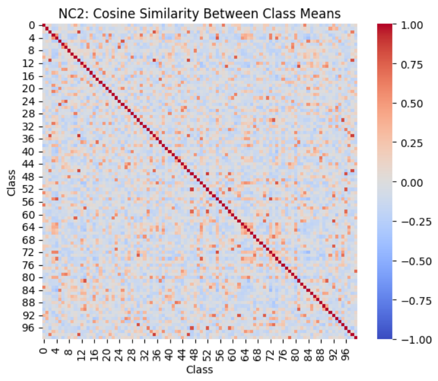

Figure 6: Class means geometry (NC2)

Here we see that NC3 is not fully reached, which is logic as we are not in a specific over-parametrised case.

Assembling the analysis previously made on NC, we can compute the OOD metric NECO. It gives the following results :

NECO performs really well on far OOD, but poorly on near OOD (slighlty less than a random classifier). Probably because neural collapse is not fully reached.

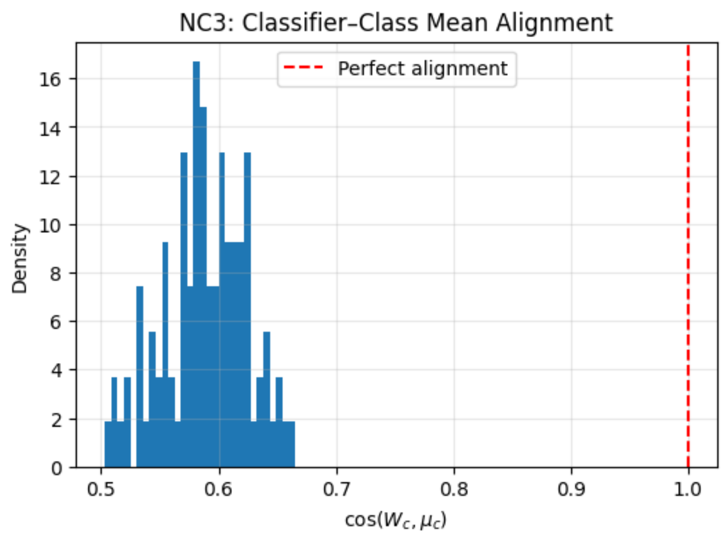

Figure 7: Learning curves

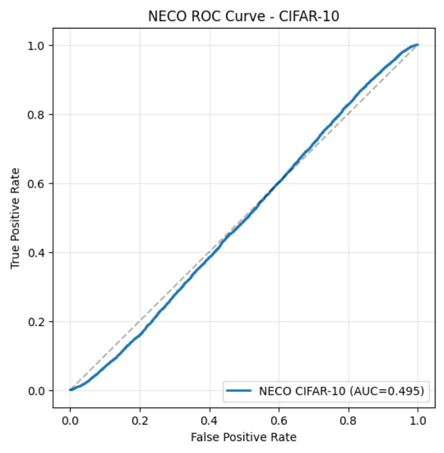

Figure 8: ROC AUC curve neco on CIFAR 10

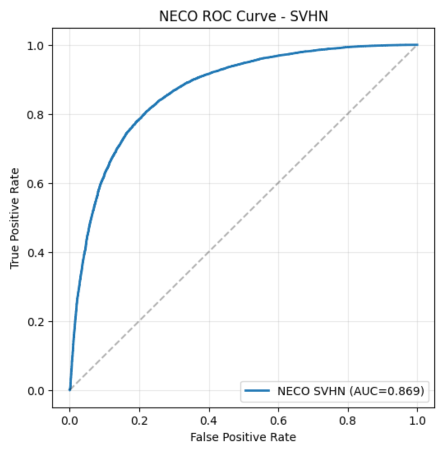

Figure 9: ROC AUC curve neco on SVHN 10