The Mimo architecture: estimating predictive uncertainty through independent subnetworks

Lorenzo Mazza **Fernando Gastón** **Anass Elyasini**

Abstract

In this project we have studied the Multi-Input-Multi-Output(MIMO) architecture in Deep Networks. The architecture was tested in several image recognition benchmarks such as Cifar10 and Cifar 100, and also on a more complex dataset that contained chest X-rays of COVID-19 patients. An additional dataset we used was the speech commands dataset, to test a different type of data besides images. We also ran some experiments on the trained networks such as testing the independence of the subnetworks.

1 Introduction

Uncertainty estimation and out-of-distribution detection play a pivotal role in the reduction of uncertainties in many applications of Deep Learning. A Deep Learning model that gives a point estimate prediction is not useful because it does not give any information on how confident the model is about that relative outcome and how far that value is from all the possible outcomes of the model. In fields where a wrong prediction can be crucial, as in the medical field, there is a highly motivated need for a quantifiable confidence for the predictions of a model. The two most common methods adopted to obtain this uncertainty estimation are Bayesian Learning and Ensemble methods, however these methods either require an excessive complexity in their notation or a high amount of computational resources, thus they are very expensive and time costly. Havasi et al. proposed in the paper "Training independent subnetworks for robust prediction" (1) a new baseline for Deep Ensembles, a single Deep Network architecture that contains M different independent subnetworks in itself, and thus it is able to exploit ensemble learning advantages in a single forward pass. As a result the network gives an estimate of the uncertainty of the model averaging the M predictions of the independent subnetworks. In this project, we replicated the architecture of the paper using a modified MIMO version of the Wide Residual Network 28-10 (2) and experiment with different datasets. We see that the MIMO model is able to give an estimate of the confidence in its predictions, so it constitutes an important step forward in the field of Uncertainty Estimation.

2 Related Work

There are several state-of-the-art works that have experimented with multi inputs or heads by modifying an existing architecture or combining several ensemble methods. Lee et al. have worked with different ensembles of Convolutional Neural Networks in (3): their idea, similarly to (**?**), is to average ensemble predictions so that the output becomes more consistent and closer to the true underlined distribution. However the architecture differs from MIMO because the different ensembles share the inputs and the weights of the first layers but the last layers and the outputs are separate, so the model is overparametrized compared to MIMO. (3) shows experiments with different model architectures besides Wide Residual Networks, and proves that the combined ensemble predictions outperformed the single model in different datasets.
Another paper that deals with reducing the computation expenses of ensemble methods is (4). The authors propose a distillation of different ensembles by distilling ensemble methods into a single multi-headed neural network by capturing only the average predictions of the different ensembles.

DD2424 - Deep Learning in Data Science (KTH 2021), Stockholm, Sweden.

This distillation does not preserve the uncertainty estimation capability of the original ensembles but has a significant gain in the computation time and memory efficiency. Hydra have outperformed different methods such as prior networks and knowledge distillation, however in some cases it performs better than MIMO and in some other cases it does not.

3 Methods

Deep residual networks were presented in (5) as a powerful evolution to deep convolutional networks, because of their capability of reaching the accuracy performance of deep networks without incurring into the degradation problem of stacking too many layers together. The residual blocks in figure 1 are the core structure of the network.

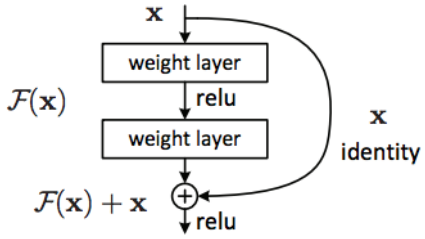

Figure 1: Residual Network Block

A wide residual network (WRN) is an architecture proposed in (2) as an improvement to deep residual networks. Compared to the latter ones, WRNs use the same residual blocks but they reduce the depth and increase the width of the network, resulting in a significant increment in terms of performance and convergence speed. The MIMO algorithm was implemented in (1) using a WRN so we implemented the same architecture. The MIMO architecture consists in a Deep CNN with M different inputs and M different outputs. At training time all the inputs are independent and belong to different classes, so the M subnetworks are trained independently. At test time the network receives M copies of the same input and outputs M independent predictions for it.

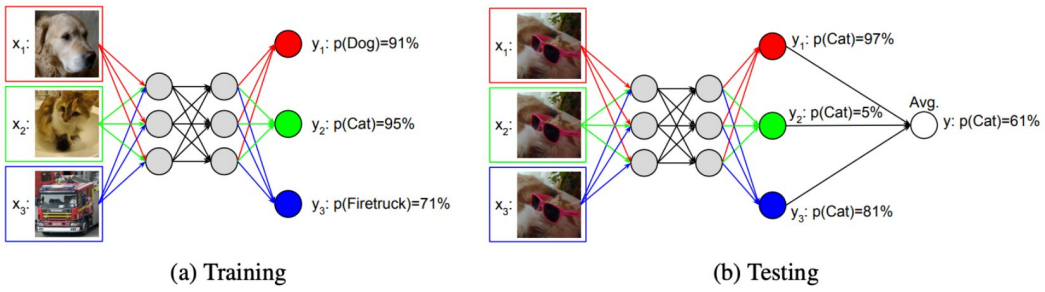

Figure 2: MIMO configuration

The code project is provided in github website: https://github.com/Lorenzo-Mazza/DD2412project

4 Data

We experimented our MIMO WRN with several different kinds of datasets; We started by training the model on Cifar10 and Cifar100 like in the original paper. Then we tried to use a different, more challenging and topical dataset, the Covid X-rays images dataset. Eventually we even tried to apply the MIMO algorith to the tensorflow speech command dataset, to test our model even with a different type of data besides images. The Cifar10, Cifar100 and speech command datasets are uploaded from the library $tensorflow\_datasets$, while the Covid x-rays dataset is taken by (6).

Cifar10 dataset consist of 60000 32x32 colour images in 10 classes, where each class consists of 6000 images. This dataset is already divided into training and testing images, where 50000 images of

training set, and 10000 images of testing set. An example for classes are; airplane, bird, dog, ship, ect. Cifar100 dataset have similar size as Cifar10, however it consists of 100 classes, and each class consists of 600 images. The training and testing set are divide similar as Cifar10.

Covid X-rays dataset consists of four different classes; COVID, Lung Opacity, Normal, and Viral Pneumonia.This dataset consist of 3616 images for COVID class, 10192 images for Normal class, 6012 images for Lung Opacity class, and 1345 images Viral Pneumonia class.

Speech command dataset consist of training, testing and validation set, our training set consists of 10% (approximate 8 thousands) of the whole training dataset (mainly for computational reasons), and testing set consists of 60% (approximate 2500) of the whole dataset.

5 Experiments and findings

5.1 Experimental settings

For our experiments, we used a WRN-28-10 for every dataset in accordance with the original paper. The 28-2 signifies that the network has a depth of $(28-4)//6 = 4$ main blocks and a widening factor of 2, which results in approximately 36.5 million trainable variables. We trained the model for 250 epochs using a decaying learning rate. The base learning value used in the paper was 0.1 times the batch size divided by 128. That resulted in a pretty big learning rate for us, so eventually we decided to test different values including 0.01, 0.001, 0.005. The learning rate had 1 warm-up epoch and three different decaying epochs over time (80th, 160th and 180th). The decay ratio was kept the same as the baseline value of 0.2. We applied different forms of regularization to the network: the first attempt was to use only L2 regularization with a value of 0.0003 like in the original paper. Then when we noticed some overfitting problems we increased the value several times in a range from 0.0003 to 0.03. We also tried to apply a combined L1-L2 regularization but we didn't achieve any significant increase in the performance so we eventually decided to discard the extra L1 regularization. Compared to the original paper although we opted neither to apply a repetition coefficient on the batches, nor to apply during training a probabilty coefficient of input repetition between the M inputs.

5.2 Performance of the MIMO architecture in image classification tasks

The metrics used in evaluating the performance were the Negative Log-Likelihood, the Accuracy and the Expected Calibration Error (7). The models were trained using the paper parameters, however several values were varied over time to test different conditions compared to the ones used by the authors.

Figure 3 shows the negative log likelihood for different datasets, the baseline number of training epochs is 250 like in the original paper, however in some runs the model training was stopped because it was implemented the Early Stopping technique, monitoring negative log-likelihood with a patience of 20 epochs. Figure 4 shows the expected calibration error of three different datasets. As we can see the final MIMO networks are well calibrated in their predictions. Figure 5 shows the training and testing accuracy for different datasets. For both Cifar10 and Cifar100 the training accuracy was really high, which means that the model overfitted the training data, For the Covid X-rays dataset the training and testing accuracies were closer in approximately all the epochs: an explanation for this is that the X-rays images had a higher level of complexity and were not overfitting as much as the Cifar ones inside the subnetworks.

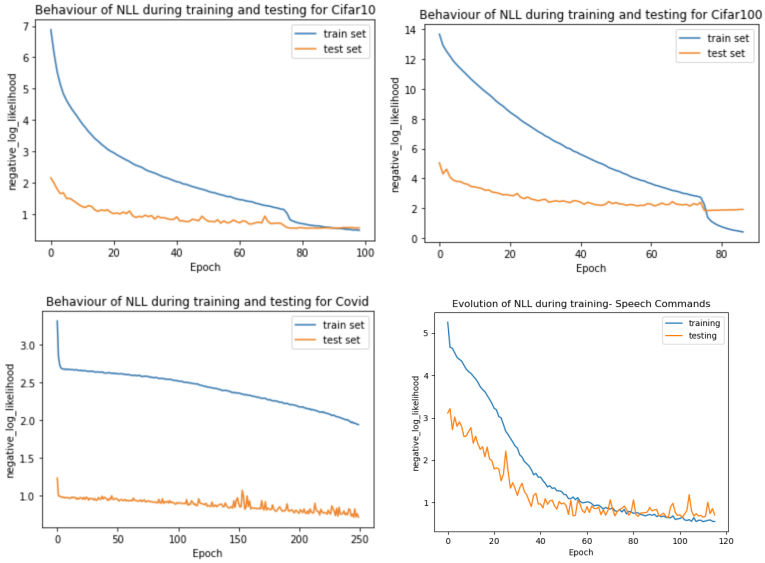

Figure 3: The negative log likelihood for Cifar10, Cifar100, Covid x-rays, Speech Commands

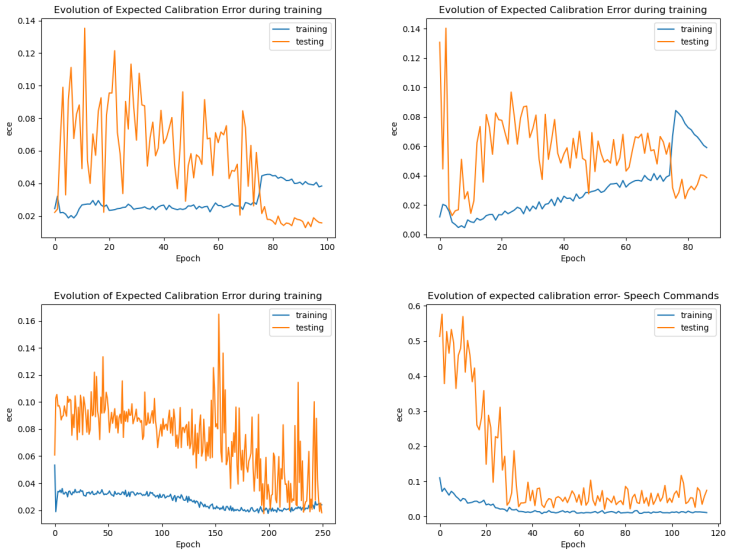

Figure 4: The Expected Calibration Error for Cifar10, Cifar100, Covid x-rays, Speech Commands

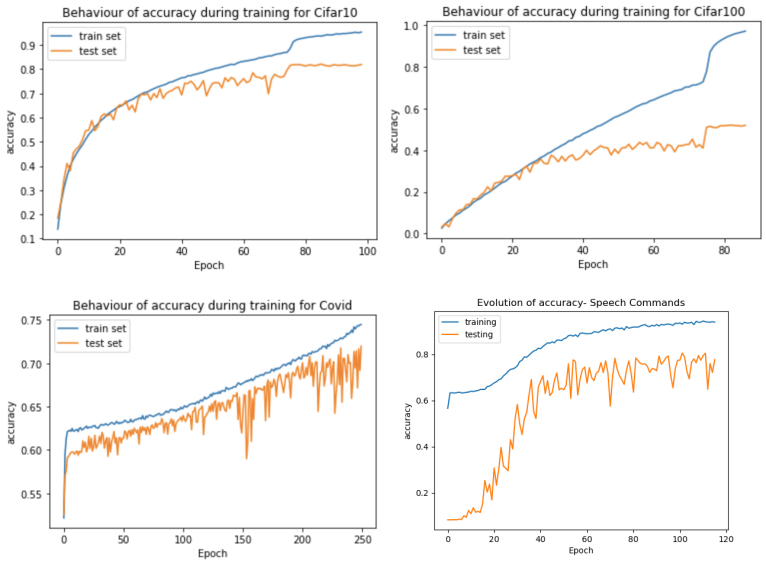

Figure 5: The accuracy for Cifar10, Cifar100, Covid x-rays, Speech Commands

5.3 Test metrics with different parameters settings

Dataset	M	$BatchSize$	$BaseLR$	$l2reg$	ece	Best NLL	Best Acc
CIFAR-10	3	256	0.1	3e-4	0.015	0.615	83.1 %
CIFAR-10	2	256	0.1	3e-4	0.018	0.685	81.4 %
CIFAR-10	3	128	0.1	3e-4	0.015	0.542	82.7 %
CIFAR-10	3	256	0.005	8e-3	0.023	0.49	88.6%
CIFAR-10	1	256	0.01	3e-4	0.030	0.287	90.8 %
CIFAR-100	3	128	0.01	3e-4	0.0127	1.842	52.3 %
COVID	3	64	0.01	3e-4	0.019	0.390	89.0 %
COVID	1	64	0.01	3e-4	0.036	0.346	89.1 %
SPEECH COMMANDS	3	16	0.01	3e-4	0.011	0.64	80.6 %

Table 1: Test accuracy for different parameter settings.

The table shows the best runs achieved with different settings of parameters. The error rates we achieve for our experiments are significantly higher than those advertised in the MIMO paper. This is also true even when we used more regularization. Fairly good accuracies were however achieved for some settings, and the Expected Calibration Error seem to behave as expected, so the MIMO model performs better than the same model with M=1. A likely explanation to this could be that we could not train the model with the orginal batch size of 512 samples because we occured in Out Of Memory errors, so we couldn't reach the same accuracy that MIMO achieves in the paper. An observation one can make is that the training loss seems to approach zero in all experiments, which might indicate that the model overfits to the training data. It is however unclear what would be the cause of this, given the fact that the same amount of regularization was used. Actually when we tried to vary the baseline learning rate and l2 value we achieved far better accuracy and NLL, so that might indicate that the values indicated by the authors are appropriated only with a Batch size of 512 samples.

5.4 Subnetwork independence

One of the main characteristics of the networks trained using the MIMO architecture is that they use disjoint parts of the network to compute the ouput for every input, giving rise to independent subnetworks. During the training phase, the inputs are sampled randomly from the training set so the network does not need to take into account the features extracted from one input to predict the output

for another one. We replicate the experiment they perform in section 3.3 to test the independence of the subnetworks where they compute the conditional variance of the pre-activations of the neurons in the network with respect to each input. For input x_1 and $M = 3$:

$$Var(a_i|x_2, x_3) = \mathop{\mathbb{E}}_{x_2,x_3}\left[\mathop{Var}_{x_1}(a_i(x_1, x_2, x_3))\right]$$

where $a_i(x_1, x_2, x_3)$ denoted the activation of the ith neuron in the network when given inputs x_1, x_2 and x_3. Figure 6 shows the results of our experiments which we run for the activations of one of the layers of a network we trained on Cifar10.

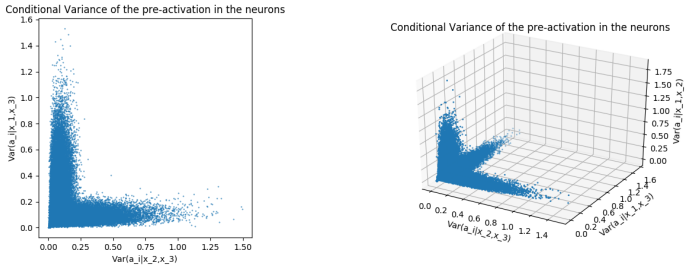

Figure 6: Conditional variance of the pre-activation of the neurons in a network with MIMO architecture trained on Cifar10 for 250 epochs.

As you can clearly see, the activation of a neuron mainly changes when modifying a particular input and not all of them. That is, the conditional variance is only big with respect to one of the inputs, which is given by the characteristic "L" shape of the plots both in the 2D plot, as well as the 3D plot. We also wanted to see how the independence of the subnetworks evolved with time. To this end, we do the same analysis on the weights we save during training. In figure 7 you can see how the independence of the subnetworks is learnt along the training and the network learns to predict the M inputs independently.

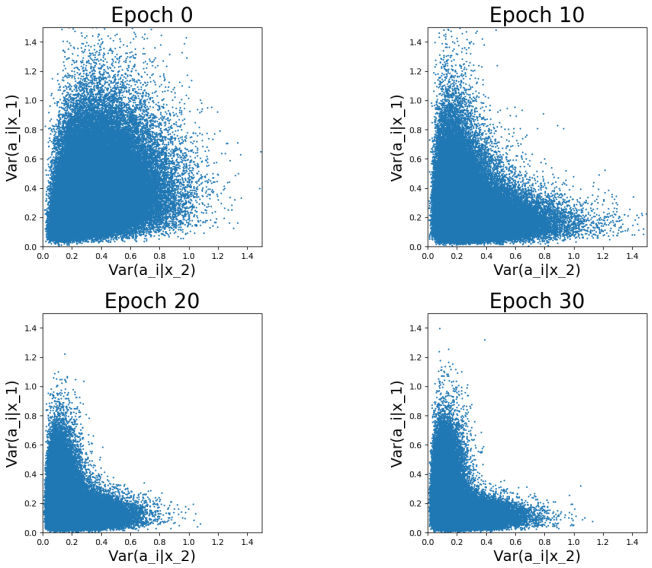

Figure 7: Subnetwork independence through the training of the network

6 Challenges

We have tried to experiment with all the datasets we planned to besides ImageNet, however some of the datasets are very large and we experimented lack of GPU. The authors trained the MIMO model on a distributed system with 4 high-quality GPUs and even TPUs in some runs. However we only could have a single 8 Gb GPU per person because of the Google Cloud GPU quota limit, so we could not implement the same hyperparameters of the original paper. As mentioned before the Batch Size had to be reduced and also the batch repetition value (how many times a single image is repeated in a training batch) could not be reproduced. The original value is 4 in the original paper but the training with it took more than 1000 seconds per epoch, thus we only experimented with batch repetition value=1. We also had problems with the datasets dimensions because of our lack of computation: we could not train the ImageNet dataset with a decent image size so eventually we decided to discard the experiments with that dataset. The Speech command dataset has almost 100000 audio files and they are very heavy, so we were forced to use only 10% of the training set, so that clearly impacted the performance of the model. In the COVID Classification dataset we had to resize the images from 256x256 to 32x32, resulting in a drop of the model accuracy.

7 Conclusion

In the project we experimented on the MIMO architecture with different configurations and different datasets. The COVID X-rays dataset was an interesting and challenging application of a real-world, trending problem so it was a very instructive experience for our future projects. The Speech commands dataset was an insightful experience because we had to transform audio files in numerical arrays using Fourier transforms. Although the MIMO model for some settings achieved rather impressive results, the accuracy rate never went beyond 90 % in Cifar10 and beyond 60 % for Cifar100, even when using the exact same configuration of the original paper. We were thus not able to replicate the results of Havasi et al. It is hard to speculate as to what exactly this depends on, and can likely depend on design flaws in our implementation. However, topic for further investigation could be the effect that Batch Size and Batch Repetition values have in implementing a MIMO architecture. For future works we suggest to experiment with different WRNs and see how they influence the independence of the subnetworks. An interesting extent would be to study the number of M different subnetworks that the network can fit without a drop in the performance in relation with the network width. Another quick improvement that could be used to improve the performance in our model would be to apply some kind of data augmentation to the inputs in order to decrease the chances for the model to overfit.

8 Self-assessment

All of us have acquires great skills and understanding of the MIMO architecture. As shown in the report we have achieved a solid theoretical and practical understanding of the algorithm even though we did not manage to exactly replicate the paper due to the reasons we listed before.

- The report is well written with relevant sources and related work which indicate that we exhaustively investigated the MIMO architecture. In addition we had a good theoretical background, different kinds of experiments and a solid discussion.
- We implemented the original MIMO algorithm writing a WRN 28-2 from scratch using Tensorflow and Keras.
- We investigated different parameters and settings for the MIMO model, considering also different values from the original ones because we did not have the computational resources of the original authors.
- We were able to test two different extra datasets that are not present in the original paper: a trending, real-world topic like COVID X-rays classification and an audio dataset, that had to be processed with complex mathematical tools to be used.
- We investigated the most complex, theoretical aspects of the original paper, analyzing the statistical independence of the subnetworks in depth, obtaining excellent results.
- With the motivations in this document we think that we have showed that we have achieved all of the Intended learning outcomes of this course.

References

[1] M. Havasi, R. Jenatton, S. Fort, J. Z. Liu, J. Snoek, B. Lakshminarayanan, A. M. Dai, and D. Tran, "Training independent subnetworks for robust prediction," *arXiv preprint arXiv:2010.06610*, 2020.

[2] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[3] S. Lee, S. Purushwalkam, M. Cogswell, D. Crandall, and D. Batra, "Why m heads are better than one: Training a diverse ensemble of deep networks," 2015.

[4] L. Tran, B. S. Veeling, K. Roth, J. Swiatkowski, J. V. Dillon, J. Snoek, S. Mandt, T. Salimans, S. Nowozin, and R. Jenatton, "Hydra: Preserving ensemble diversity for model distillation," 2020.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[6] M. E. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, K. R. Islam, M. S. Khan, A. Iqbal, N. Al Emadi *et al.*, "Can ai help in screening viral and covid-19 pneumonia?" *IEEE Access*, vol. 8, pp. 132 665–132 676, 2020.

[7] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.