IDATG2208 - Mandatory Assignment - 3

Deadline for submission - 05 Nov, 2025

Instructions

- Set random seeds to 42 for all experiments.
- Use only the specified dataset subset for reproducibility.
- Report your answers as percent numbers for test accuracy. The code and plots depicting training and validation loss should be provided along with the test accuracy.

Exercise-1: Training Deep Neural Network on MNIST [15 points]

Train a controlled deep neural network on the MNIST dataset. Set random seeds to 42. Load and preprocess MNIST. Build the network using the following configuration:

- Flatten input images to $28 \times 28 = 784$ features
- 3 hidden layers, 64 neurons each
- ELU activation function
- He normal initialization
- Output layer: 10 neurons with softmax
- Optimizer: Nadam
- $learning\_rate = 0.001$, loss=`sparse_categorical_crossentropy`
- EarlyStopping callback: monitor validation loss, $patience = 5$, restore best weights
- $epochs = 50$, $batch\_size = 32$
- Use only the first 1000 training samples and first 200 test samples

Q1.1 Report the obtained test accuracy.

Exercise-2: Training Deep Neural Network on CIFAR-10 [15 points]

Train a controlled deep neural network on the CIFAR-10 dataset. Set random seeds to 42. Load and preprocess CIFAR-10. Build the network using the following configuration:

- Flatten input images to $32 \times 32 \times 3 = 3072$ features
- 4 hidden layers, 256 neurons each
- ELU activation function
- He normal initialization
- Output layer: 10 neurons with softmax
- Optimizer: Nadam
- $learning\_rate = 0.001$, $loss =' sparse\_categorical\_crossentropy'$
- EarlyStopping callback: monitor validation loss, $patience = 5$, restore best weights
- $epochs = 50$, $batch\_size = 128$
- Use only the first 5000 training samples and first 1000 test samples

Q2.1 Report the obtained test accuracy.

Exercise-3: Regularization with Alpha Dropout and MC Dropout [15 points]

Using the MNIST dataset, extend the previously trained deep neural network by applying Alpha Dropout. Then, without retraining, use Monte Carlo (MC) Dropout at inference to estimate if you can achieve better accuracy. Set random seeds to 42. Use the following configuration:

- Flatten input images to $28 \times 28 = 784$ features
- 3 hidden layers, 64 neurons each
- SELU activation function (required for Alpha Dropout)
- LeCun normal initialization
- Alpha Dropout rate: 0.1 in all hidden layers
- Output layer: 10 neurons with softmax
- Optimizer: Nadam
- $learning\_rate = 0.001$, loss=`sparse_categorical_crossentropy`
- $epochs = 50$, $batch\_size = 32$

- Use only the first 1000 training samples and first 200 test samples
- For MC Dropout, enable dropout during inference and average predictions over 20 stochastic forward passes

Q3.1 Report the test accuracy of the network with Alpha Dropout applied during training.

Q3.2 Report the MC Dropout-enhanced accuracy (averaging 20 stochastic predictions).

Exercise-4: Transfer Learning with Pre-trained CNN [15 points]

Use a pre-trained convolutional neural network (CNN) as a feature extractor and fine-tune a classifier on a subset of the CIFAR-10 dataset. Set random seeds to 42. Follow the configuration below:

- Load CIFAR-10 and normalize pixel values to [0,1]
- Use only the first 2000 training samples and first 500 test samples
- Load `MobileNetV2` from `tensorflow.keras.applications`, with `include_top=False` and `weights='imagenet'`
- Freeze all layers of the pre-trained base
- Add a classifier on top:
  - GlobalAveragePooling2D
  - Dense layer with 128 neurons, ReLU activation
  - Dropout: 0.2
  - Output layer: 10 neurons with softmax
- Optimizer: Adam, $learning\_rate = 0.001$
- Loss: `sparse_categorical_crossentropy`
- $epochs = 5$, $batch\_size = 32$

Q4.1 Report the test accuracy of the model.

Exercise-5: Deeper CNN Training on SVHN [20 points]

Train a controlled deep convolutional neural network (CNN) on a subset of the SVHN dataset. Set random seeds to 42. Load and preprocess SVHN. Build the network using the following configuration:

- Load SVHN and normalize pixel values to [0,1]

- Use only the first 2000 training samples and first 500 test samples
- Input shape: $32 \times 32 \times 3$
- CNN architecture:
  - Conv2D: 32 filters, 3×3 kernel, ReLU activation
  - Conv2D: 32 filters, 3×3 kernel, ReLU activation
  - MaxPooling2D: 2×2
  - Conv2D: 64 filters, 3×3 kernel, ReLU activation
  - Conv2D: 64 filters, 3×3 kernel, ReLU activation
  - MaxPooling2D: 2×2
  - Flatten
  - Dense: 256 neurons, ReLU activation
  - Dropout: 0.3
  - Output layer: 10 neurons with softmax
- Optimizer: Adam, $learning\_rate = 0.001$
- Loss: `sparse_categorical_crossentropy`
- $epochs = 15$, $batch\_size = 32$

Q5.1 Report the test accuracy of the deeper CNN model.

Exercise-6: CNN with SGD, MC Dropout, and Epistemic Uncertainty [20 points]

Train a controlled convolutional neural network (CNN) on a subset of the SVHN dataset using SGD optimizer. Then, apply Monte Carlo (MC) Dropout at inference to estimate both test accuracy and epistemic uncertainty. Set random seeds to 42. Use the following configuration:

- Load SVHN and normalize pixel values to [0,1]
- Use only the first 2000 training samples and first 500 test samples
- Input shape: $32 \times 32 \times 3$
- CNN architecture:
  - Conv2D: 32 filters, 3×3 kernel, ReLU activation
  - MaxPooling2D: 2×2
  - Conv2D: 64 filters, 3×3 kernel, ReLU activation
  - MaxPooling2D: 2×2
  - Flatten

- Dense: 128 neurons, ReLU activation
  - Dropout: 0.25 (keep during inference for MC Dropout)
  - Output layer: 10 neurons with softmax

- Optimizer: SGD with momentum = 0.9, $learning\_rate = 0.01$
- Loss: `sparse_categorical_crossentropy`
- $epochs = 15$, $batch\_size = 32$
- For MC Dropout:
  - Enable dropout during inference
  - Average predictions over 20 stochastic forward passes
  - Compute the epistemic uncertainty as the predictive variance across passes

Q6.1 Report the plain test accuracy of the CNN trained with SGD (no MC Dropout).

Q6.2 Report the MC Dropout-enhanced accuracy (averaging 20 stochastic predictions).

Q6.3 Compute the average epistemic uncertainty (mean predictive variance) across all test samples. Report it as a deterministic number rounded to 3 decimal places.