Supplementary Materials

Table S1 presents the full experimental test of the Validation accuracy measurements. The table elaborates on the analysis below, which is presented in the paper as Table 3. Validation accuracy improvement, average, and maximum.

As described below, in some cases, more than one experiment was run per dataset, such as for small datasets that allow running without overhead and complexity (which are out of scope for this paper). For example, in Table S1, index 2, the dataset Adult (full references are in the paper) was run 5 times. In case of multiple runs, the average of all runs were used in the calculation, as shown in the column "Validation accuracy improvement average" in Table 4 in the paper, and the max results of all the tests per dataset were noted in the column "Validation accuracy max improvement" in Table S3. The full table (Table S1) was used for the statistical analysis as well

Table S1. Full data output – Baseline vs. Model - Validation accuracy.

dataset_index/dataset name	baseliine_val_acc	val_acc_model
1/Abalone	0.233254	0.295455
1/Abalone	0.26555	0.295455
2/Adult	0.835876	0.839536
2/Adult	0.844417	0.848078
2/Adult	0.80781	0.822453
2/Adult	0.850519	0.85723
2/Adult	0.849298	0.854179
3/Balance Scale	0.944	0.976
3/Balance Scale	0.896	0.904
3/Balance Scale	0.944	0.984
3/Balance Scale	0.944	0.984
3/Balance Scale	0.936	0.96
4/Bank Marketing	0.956071	0.971324
4/Bank Marketing	0.968883	0.971324
4/Bank Marketing	0.967663	0.968883
5/Breast Cancer	0.754386	0.807018
5/Breast Cancer	0.719298	0.754386
5/Breast Cancer	0.719298	0.754386
5/Breast Cancer	0.77193	0.824561
5/Breast Cancer	0.789474	0.824561
6/car evaluation	0.988439	0.991329
7/CDC Diabetes Health Indicators	0.835265	0.837096
7/CDC Diabetes Health Indicators	0.826724	0.844417
7/CDC Diabetes Health Indicators	0.837096	0.844417
8/Census Income	0.835876	0.839536

8/Census Income	0.844417	0.848078
8/Census Income	0.80781	0.822453
8/Census Income	0.850519	0.85723
8/Census Income	0.849298	0.854179
9/Default of Credit Card Clients	0.710189	0.8072
9/Default of Credit Card Clients	0.790726	0.8072
9/Default of Credit Card Clients	0.788896	0.801098
9/Default of Credit Card Clients	0.796217	0.801098
10/Dry Bean	0.958511	0.962782
10/Dry Bean	0.964002	0.964613
10/Dry Bean	0.965223	0.967053
10/Dry Bean	0.962782	0.967053
11/Estimation of Obesity Levels Based On Eating Habits and Physical Condition	0.869668	0.912322
11/Estimation of Obesity Levels Based On Eating Habits and Physical Condition	0.921801	0.933649
11/Estimation of Obesity Levels Based On Eating Habits and Physical Condition	0.952607	0.971564
11/Estimation of Obesity Levels Based On Eating Habits and Physical Condition	0.957346	0.964455
12/Letter Recognition	0.816351	0.818792
13/National Poll on Healthy Aging	0.482517	0.524476
13/National Poll on Healthy Aging	0.503497	0.524476
13/National Poll on Healthy Aging	0.461538	0.475524
14/Nursery	0.914582	0.928615
14/Nursery	0.907261	0.913362
14/Nursery	0.99817	1
14/Nursery	0.981696	0.983527
15/Online News Popularity	0.053081	0.053691
15/Online News Popularity	0.04881	0.054301
15/Online News Popularity	0.018304	0.021965
15/Online News Popularity	0.039048	0.059182
15/Online News Popularity	0.04881	0.059182
16/Online Shoppers Purchasing Intention Dataset	0.916412	0.921293
16/Online Shoppers Purchasing Intention Dataset	0.917023	0.919463
16/Online Shoppers Purchasing Intention Dataset	0.90482	0.910921
17/Statlog	0.69	0.76
17/Statlog	0.725	0.76
17/Statlog	0.645	0.755
17/Statlog	0.735	0.755
17/Statlog	0.725	0.745
18/student dropout	0.740113	0.745763

18/student dropout	0.751412	0.760452
19/Toxicity	0.588235	0.617647
19/Toxicity	0.588235	0.647059
20/wine quality	0.543846	0.562308
20/wine quality	0.582308	0.589231
20/wine quality	0.561538	0.584615
20/wine quality	0.573077	0.579231

Below Table S2 index lists key variables used in the article Table 2. Detailed method procedure steps. Each variable includes its type, description, and typical values or usage context, facilitating elaboration and understanding for scientific research.

Note that refers for both Table S2 and Table S3 below, and for this paper: while during the research experiment, we used PyTorch [1] library functions, the implementation (coding language, environment, data, etc.) is out of scope for the research, and therefore, additional modifications and adjustments may be needed for coding, language, environment, and implementation.

Table S2. index lists key variables

Variable	Type	Description and Usage
Inputs to Procedure:		
DatasetName	String	Name of the UCI dataset to load, e.g., "wine quality" or "mushroom". Determines data fetch.
ModelType	Enum	Specifies model architecture: SimpleNN for single-layer neural net, Classifier for multi-layer with dropout, TextClassificationModelCnn for convolutional neural net. Affects instantiation.
input_dim	Integer	Model input dimension
num_classes	Integer	Number of classes that the model predicts
Config.HIDDEN_DIM	Integer	The hidden dimension used in the model
Config.DROPOUT, dropout_rate	Float	Dropout rate for each layer in the model
Config.training_ratio	Float	Variable to split the training data between training and validation
UseScheduler	Boolean	If true, uses a learning rate scheduler (ReduceLROnPlateau) during training; else, fixed LR.
conv_channels, kernel_sizes, dense_layers,	Integer	CNN based classifier parameters
MaxDatasetSize	Integer	Maximum allowed dataset features (default 8192); truncates larger datasets to prevent long training.
BenchmarkLoss	Float	Target validation loss threshold; training stops early if reached.

Variable	Type	Description and Usage
TargetAcc	Float	Target accuracy threshold; training stops early if reached.
Locals in Procedure:		
D	Dataset Object	Fetched UCI dataset object containing features (X) and targets (y). Truncated if too large.
M	Model Object	Instantiated neural network model (e.g., SimpleNN or Classifier). Holds weights, forward/backward logic.
C	Loss Function	Cross-entropy loss for multi-class classification, used in optimizer steps.
O	Optimizer	Optimizer object (e.g., RMSprop or Adam), configured with LR, weight decay, momentum.
S	Scheduler	Optional learning rate scheduler object; adjusts LR based on plateau in validation loss.
loss, acc, epochs	Arrays	Output from train_model: lists of training losses, accuracies per epoch, and total epochs run.
L_b	Float	Minimum validation loss from benchmark training, used as early stopping criterion.
A_b	Float	Maximum validation accuracy from benchmark training, used as early stopping criterion.
best_score	Float	Highest selection score encountered during seed search (starts at 0).
best_seed	Integer	Random seed yielding the best_score (null initially).
best_dist	Array	Best distribution configuration [homebrew_flag, RD_type, description].
dist_list	Array	One of 6 predefined distribution configurations for random initialization (e.g., [False, RD.UNIFORM]).
it	Integer	Loop counter for seed search iterations.
seed	Integer	Current random seed generated (0 to 9999).
loss_s, acc_s	Arrays	Short training outputs: validation losses and accuracies over E_s epochs.
current_score	Float	Computed selection criterion for current seed/run: $1/\left(\left(loss_s[-1] * (1 - acc_s[-1])\right) + \epsilon\right)$.
loss_f, acc_f, epochs_f	Arrays	Final training outputs after best seed selection.
Config Constants:		
EPOCHS_FOR_BEST_SEED	Integer	Short training epochs per seed (typically 2).

Variable	Type	Description and Usage
NUMBER_OF_ITERATIONS_FOR_BEST_SEED	Integer	Number of seed trials per distribution (typically 10).
RD.UNIFORM, RD.NORMAL, RD.CONSTANT	Enum	Random distribution types: uniform random values, normal (Gaussian), or constant.
Config.training_ratio	Float	Ratio for randomization strength (e.g., 0.001).
Config.NUM_EPOCHS	Integer	Maximum epochs for full training (e.g., 50).
Config.HIDDEN_DIM	Integer	Hidden layer dimensions for models like Classifier (e.g., 64).
Config.DROPOUT	Float	Dropout probability for regularization (e.g., 0.3).
ϵ	Float	Small constant (1e-8) to avoid division by zero in criterion.

Below Table S3: index functions and pseudo-functions used in the article Table 2. Detailed method procedure steps. Each function includes its description and usage. and return type. facilitating elaboration and understanding for scientific research.

Table S3. index functions and pseudo-functions

Function	Return Type	Description and Usage
Data Handling:		
fetch_ucirepo(DatasetName)	Dataset Object	Fetches UCI dataset by name. Returns features (X), targets (y), and metadata. Used in line 2 to load D.
Model Instantiation:		
SimpleNN(input_dim, num_classes)	Model Object	Initializes a simple neural network with single hidden layer and softmax output. Used for basic topology.
Classifier(input_dim, hidden_dim, num_classes, dropout)	Model Object	Initializes a multi-layer classifier with hidden dimensions, dropout, and classification layers. Relies on Config.DROPOUT for regularization.
TextClassificationModelCnn(input_dim, num_classes, conv_channels, kernel_sizes, dense_layers, dropout_rate)	Model Object	Initializes a CNN-based model for text classification with convolutional and dense layers. Suitable for higher-dimensional data.
Training Infrastructure:		
CrossEntropyLoss()	Loss Function	Creates cross-entropy loss object for multi-class tasks. Passed as C to training function.
RMSprop or Adam(params, lr=LR)	Optimizer	Initializes optimizer (RMSprop for SimpleNN/Classifier; Adam for CNN) with learning rate and weight decay. Stored as O. These functions use PyTorch library functions

ReduceLROnPlateau(optimizer, mode='min', factor=0.1, patience=5)	Scheduler	Creates LR scheduler that reduces learning rate on validation plateau. Assigned to S if UseScheduler is true.
train_model(model, train_loader, val_loader, criterion, optimizer, scheduler, max_epochs, loss_target, acc_target)	Tuple (loss, acc, epochs)	Core training loop: runs up to max_epochs, updates model weights on train_loader, validates on val_loader, stops if loss < loss_target or acc > acc_target. Returns lists of losses, accuracies, total epochs. Called twice: benchmark (line 14) and final (line 35).
Initialization:		
apply_random_function_to_weights(model, seed, distribution, training_ratio, homebrew_flag)	Void	Applies randomized weight initialization based on seed, distribution (uniform/normal/constant), and ratio. If homebrew_flag, uses custom randomization. Modifies model weights in-place for seed search/finding.
Utilities(All are python standard functions):		
random(min, max)	Integer	Generates random integer between min and max. Used for seed generation (e.g., 0 to 9999).
min(loss_array), max(acc_array)	Float	Numpy-like min/max functions for arrays. Extracts benchmark metrics ((L_b, A_b)

[1] Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv* **2019**, arXiv:1912.01703