Training independent subnetworks for robust prediction

Havasi, M., Jenatton, R., Fort, S., Liu, J. Z.,
Snoek, J., Lakshminarayanan, B., Dai, A. M., Tran, D. (2021)

citations: 102

Outline

- Recap: Explicit Ensembles
- Multi-Input Multi-Output (MIMO)
- Baselines
- Evaluation Scores
- Benchmarks
- Ablation
- Hyperparameters
- Summary

Recap: Explicit Ensembles

Benefits

Drawbacks

Recap: Explicit Ensembles

Benefits

- provide uncertainty metrics (variance of predictions)
- better calibration and generalization (diverse assumptions)

Drawbacks

Recap: Explicit Ensembles

Benefits

- provide uncertainty metrics (variance of predictions)
- better calibration and generalization (diverse assumptions)

Drawbacks

- M members are stored
- M forward passes (also in inference)

Recap: Explicit Ensembles

Benefits

- provide uncertainty metrics (variance of predictions)
- better calibrati an ation
  (dive

implicit ensembles
within one network
evade the drawbacks

Drawbacks

- M memb are stored
- M forw es (als nce)

Multi-Input Multi-Output (MIMO)

Neural Network

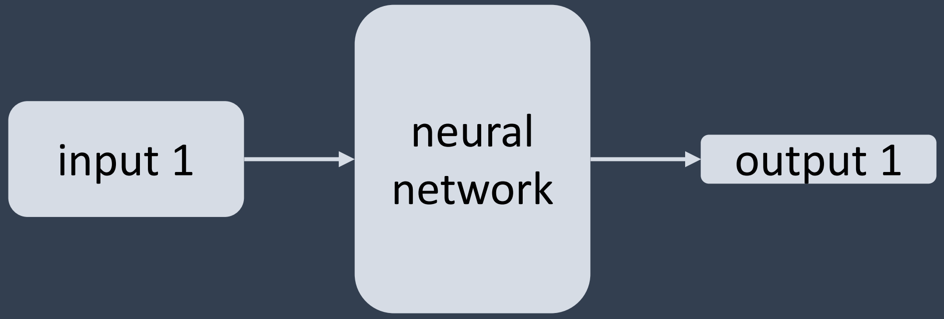

Figure 1: normal neural network

- is overparameterized

MIMO

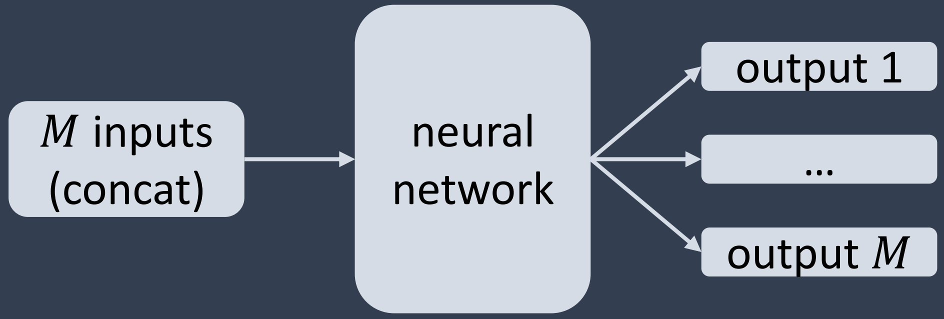

Figure 2: MIMO

- M times more channels in input layer
- M times more output heads
- + 0.01% parameters (ResNet28-10)
- + 0.03% evaluation FLOPs (ResNet28-10)

MIMO for inference

- repeat input M times
- predictions are averaged

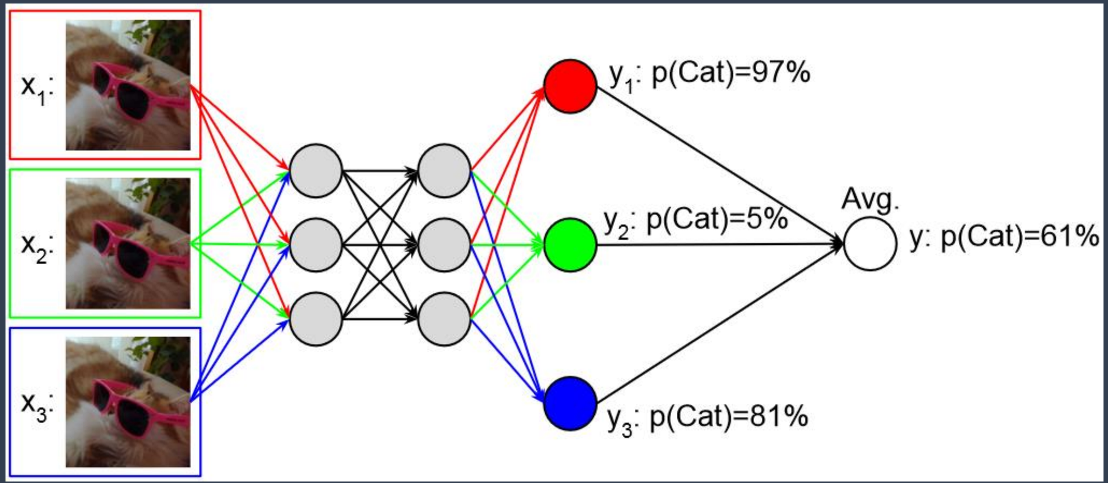

Figure 3: MIMO during inference, figure made by [1]

MIMO training

- draw M inputs independently
- head m has to predict label of input m
- loss: $\Sigma_m(\text{neg. log-likelihood})_{head_m} + R(\theta)$
- implicitly trains M subnetworks

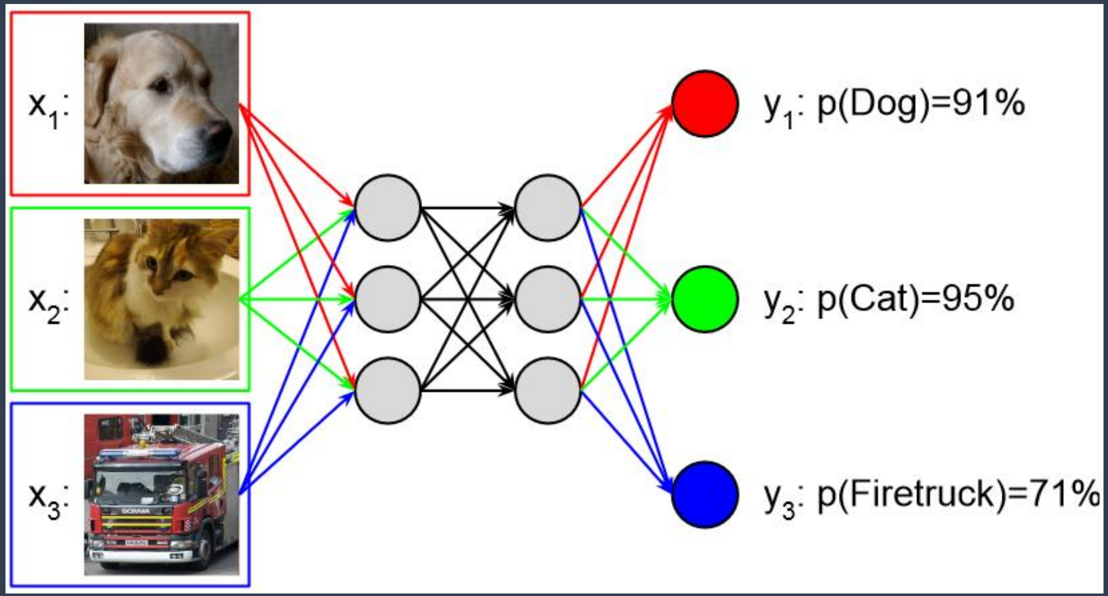

Figure 4: MIMO during training, figure made by [1]

Baselines

- (deterministic) neural network
- naive multihead
- TreeNet [3]
- deep ensemble [2]
- BatchEnsemble [4]
- Monte Carlo Dropout [5]

Baselines

- (deterministic) neural network

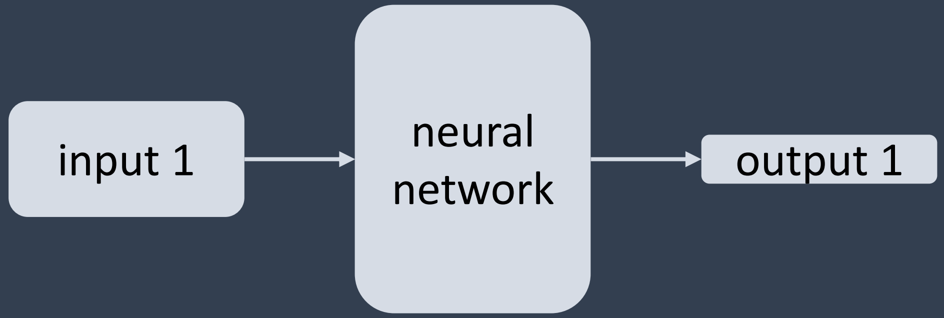

Figure 5: (deterministic) neural network

Baselines

- (deterministic) neural network
- naive multihead

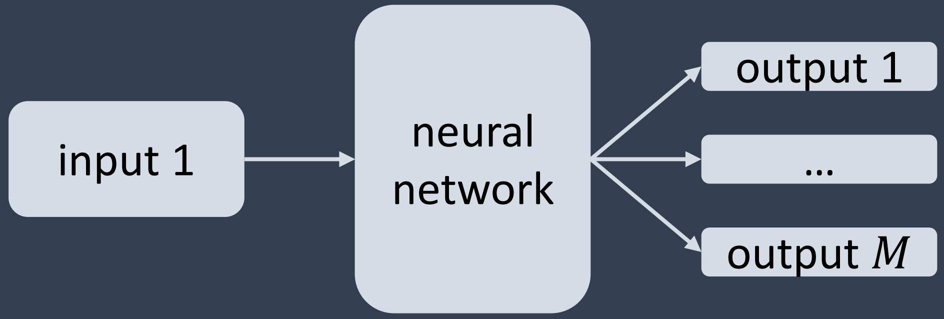

Figure 6: naive multihead

- small individual output heads

Baselines

- (deterministic) neural network
- naive multihead
- TreeNet [3]

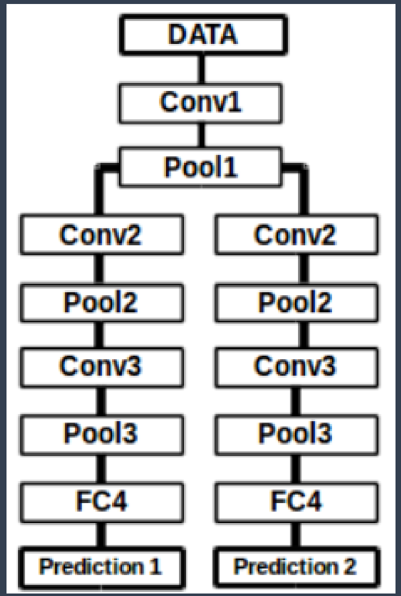

Figure 7: TreeNet, figure made by [3]

- members share few generic first layers

Baselines

- (deterministic) neural network
- naive multihead
- TreeNet [3]
- deep ensemble [2]

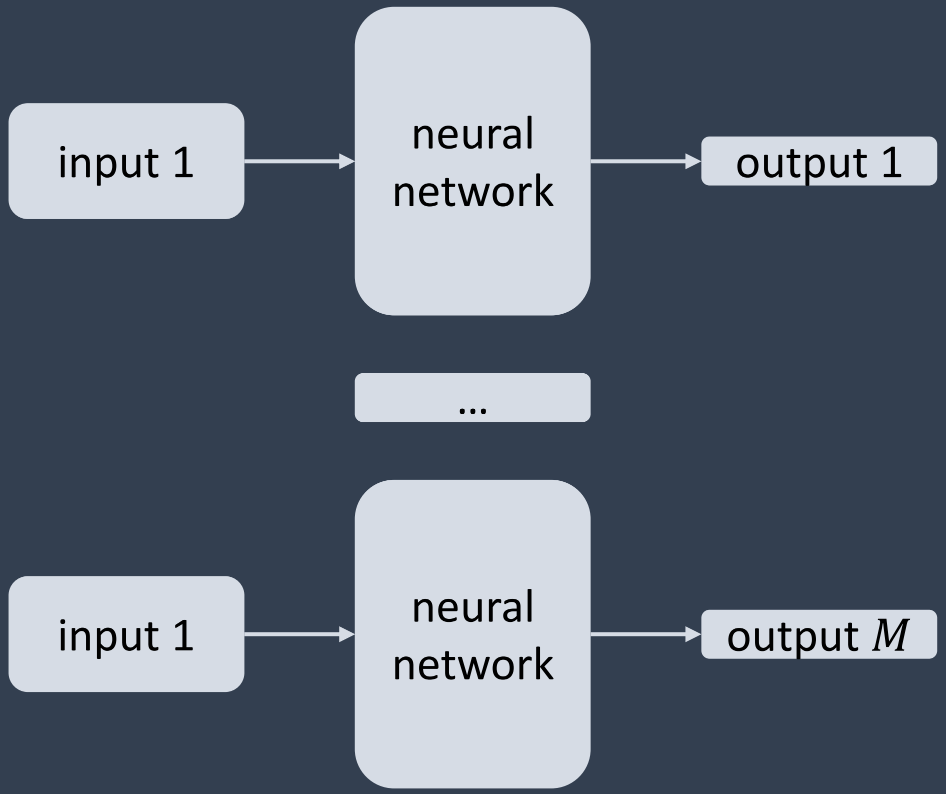

Figure 8: deep ensemble

Baselines

- (deterministic) neural network
- naive multihead
- TreeNet [3]
- deep ensemble [2]
- BatchEnsemble [4]

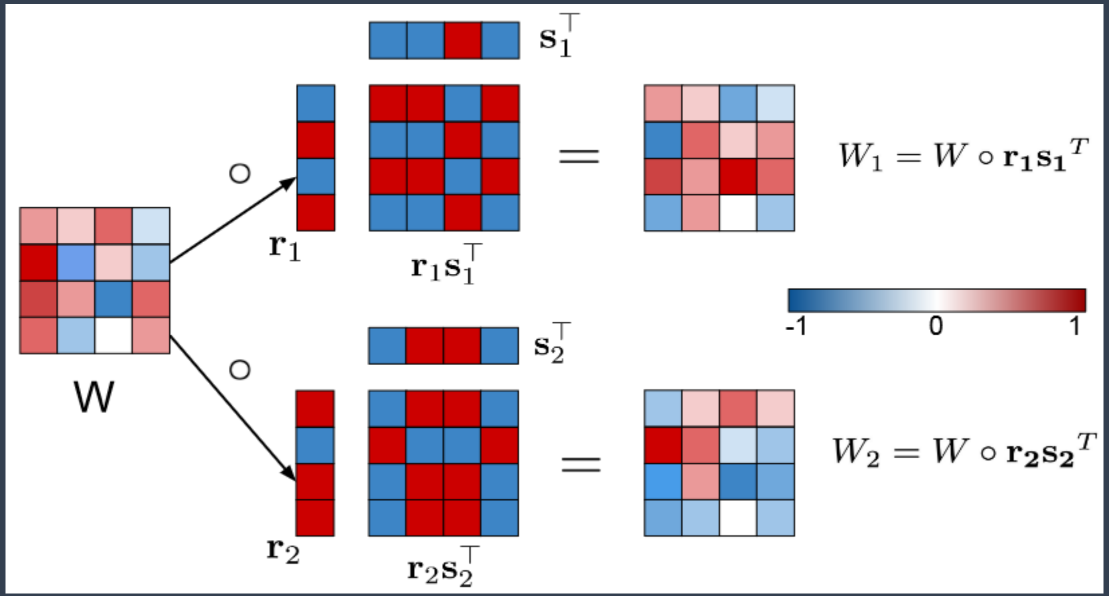

Figure 9: BatchEnsemble weights, figure made by [4]

- members have individual “fast weights”

Baselines

- (deterministic) neural network
- naive multihead
- TreeNet [3]
- deep ensemble [2]
- BatchEnsemble [4]
- Monte Carlo Dropout [5]

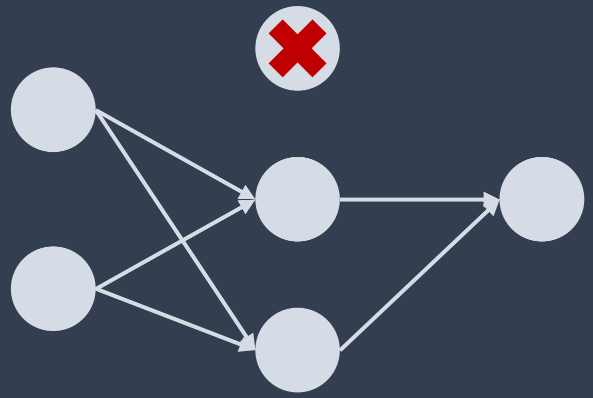

Figure 10: Dropout

- M dropout forward passes in inference

Evaluation Scores

- accuracy
- neg. log-likelihood
- expected calibration error (ECE)

ECE: a score for a models’ uncertainty awareness

ECE: a score for a models' uncertainty awareness

good model is aware of
its prediction
uncertainty

ECE: a score for a models’ uncertainty awareness

1. predict confidence on whole test set

Figure 11: ECE toy example

- green is correct prediction
- red is wrong prediction

ECE: a score for a models' uncertainty awareness

1. predict confidence on whole test set
2. bin data points by confidence

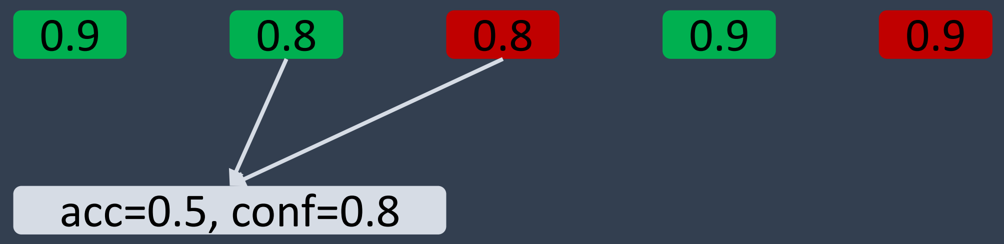

Figure 11: ECE toy example

ECE: a score for a models’ uncertainty awareness

1. predict confidence on whole test set
2. bin data points by confidence

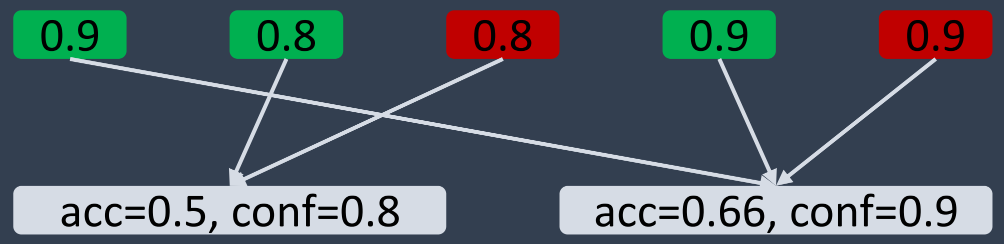

Figure 11: ECE toy example

ECE: a score for a models' uncertainty awareness

1. predict confidence on whole test set
2. bin data points by confidence
3. ECE is weighted mean of difference between acc and confidence along bins

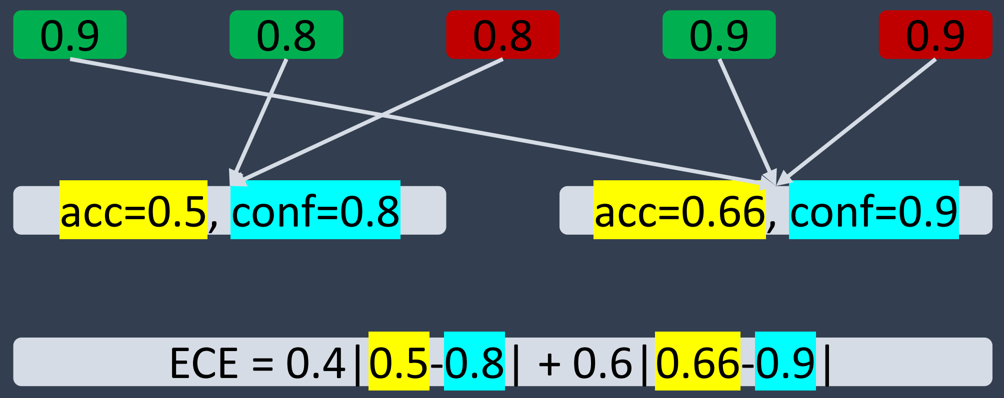

Figure 11: ECE toy example

- $\boldsymbol{ECE = \sum_b \frac{|b|}{n} |acc(b) - conf_{avg}(b)|}$

ECE: a score for a models’ uncertainty awareness

1. predict confidence on whole test set
2. bin data points by confidence
3. ECE is weight … nce
   bet
   bi

ECE calculates the difference between the confidence and the actual uncertainty

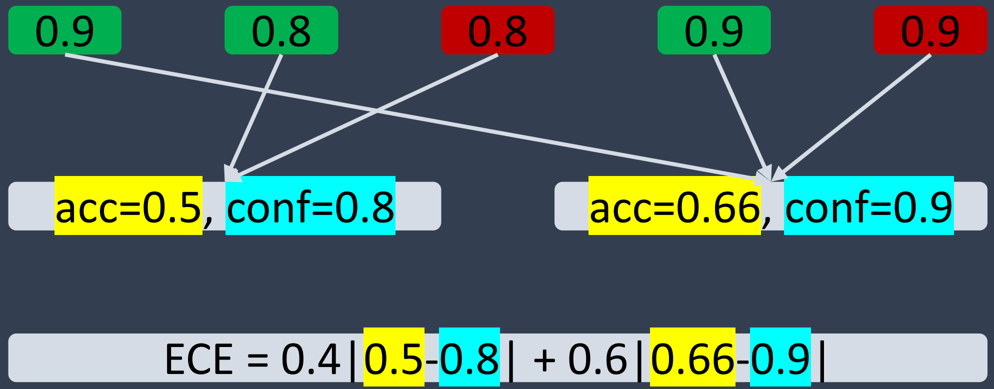

Figure 11: ECE toy example

- $ECE = \sum_b \frac{|b|}{n} |acc(b) - conf_{avg}(b)|$

Benchmarks

Name	Accuracy (↑)	NLL (↓)	ECE (↓)	cAcc (↑)	cNLL (↓)	cECE (↓)	Prediction time (↓)	# Forward passes (↓)
Deterministic	79.8	0.875	0.086	51.4	2.700	0.239	0.632	1
Monte Carlo Dropout	79.6	0.830	0.050	42.6	2.900	0.202	0.656	1
Naive mutlihead ($M = 3$)	79.5	0.834	0.048	52.1	2.339	0.156	0.636	1
MIMO ($M = 3$) (This work)	**82.0**	**0.690**	**0.022**	**53.7**	**2.284**	**0.129**	0.639	1
TreeNet ($M = 3$)	80.8	0.777	0.047	53.5	2.295	0.176	0.961	1.5
BatchEnsemble ($M = 4$)	81.5	0.740	0.056	54.1	2.490	0.191	2.552	4
Thin Ensemble ($M = 4$)	81.5	0.694	0.017	53.7	2.190	0.111	0.823	4
Ensemble ($M = 4$)	82.7	0.666	0.021	54.1	2.270	0.138	2.536	4

Table 1: Results on CIFAR10 with ResNet28-10, table made by [1]

- **perturbated test data: cAcc, cNLL, cECE**
- **prediction time in ms per data point**
- **similar on CIFAR100 with ResNet28-10 and on ImageNet with ResNet50**

Benchmarks

Name	Accuracy (↑)	NLL (↓)	ECE (↓)	cAcc (↑)	cNLL (↓)	cECE (↓)	Prediction time (↓)	# Forward passes (↓)
Deterministic	79.8	0.875	0.086	51.4	2.700	0.239	0.632	1
Monte Carlo Dropout	79.6	0.830	0.050	42.6	2.900	0.202	0.656	1
Naive mutlihead ($M = 3$)	79.5	0.834	0.048	52.1	2.339	0.156	0.636	1
MIMO ($M = 3$) (This work)	**82.0**	**0.690**	**0.022**	**53.7**	**2.284**	**0.129**	0.639	1
TreeNet ($M = 3$)	80.8	0.777	0.047	53.5	2.295	0.176	0.961	1.5
BatchEnsemble ($M = 4$)	81.5	0.740	0.056	54.1	2.490	0.191	2.552	4
Thin Ensemble ($M = 4$)	81.5	0.694	0.017	53.7	2.190	0.111	0.823	4
Ensemble ($M = 4$)	82.7	0.666	0.021	54.1	2.270	0.138	2.536	4

Table 1: Results on CIFAR10 with ResNet28-10, table made by [1]

- **perturbated test data: cAcc, cNLL, cECE**
- **prediction time in ms per data point**
- **similar on CIFAR100 with ResNet28-10 and on ImageNet with ResNet50**

Benchmarks

Name	Accuracy (↑)	NLL (↓)	ECE (↓)	cAcc (↑)	cNLL (↓)	cECE (↓)	Prediction time (↓)	# Forward passes (↓)
Deterministic	79.8	0.875	0.086	51.4	2.700	0.239	0.632	1
Monte Carlo Dropout	79.6	0.830	0.050	42.6	2.900	0.202	0.656	1
Naive mutlihead ($M = 3$)	79.5	0.834	0.048	52.1	2.339	0.156	0.636	1
MIMO ($M = 3$) (This work)	**82.0**	**0.690**	**0.022**	**53.7**	**2.284**	**0.129**	0.639	1
TreeNet ($M = 3$)	80.8	0.777	0.047	53.5	2.295	0.176	0.961	1.5
BatchEnsemble ($M = 4$)	81.5	0.740	0.056	54.1	2.490	0.191	2.552	4
Thin Ensemble ($M = 4$)	81.5	0.694	0.017	53.7	2.190	0.111	0.823	4
Ensemble ($M = 4$)	82.7	0.666	0.021	54.1	2.270	0.138	2.536	4

Table 1: Results on CIFAR10 with ResNet28-10, table made by [1]

- **perturbated test data: cAcc, cNLL, cECE**
- **prediction time in ms per data point**
- **similar on CIFAR100 with ResNet28-10 and on ImageNet with ResNet50**

Benchmarks

Name	Accuracy (↑)	NLL (↓)	ECE (↓)	cAcc (↑)	cNLL (↓)	cECE (↓)	Prediction time (↓)	# Forward passes (↓)
Deterministic	79.8	0.875	0.086	51.4	2.700	0.239	0.632	1
Monte Carlo Dropout	79.6	0.830	0.050	42.6	2.900	0.202	0.656	$\boxed{1}$
Naive mutlihead (M[illegible]	[illegible]	[illegible]	0.048	52.1	2.339	0.156	0.636	1
MIMO [illegible]	[illegible]	[illegible]	**0.022**	**53.7**	**2.284**	**0.129**	0.639	1
TreeN[illegible]	[illegible]	[illegible]	0.047	53.5	2.295	0.176	0.961	1.5
Batch[illegible]	[illegible]	[illegible]	0.056	54.1	2.490	0.191	2.552	4
T[illegible]	[illegible]	[illegible]	0.017	53.7	2.190	0.111	0.823	4
[illegible]	[illegible]	[illegible]	0.021	54.1	2.270	0.138	2.536	4

problem:
1 forward pass with
mc-dropout provides
point prediction only

…AR10 with ResNet28-10, table made by [1]

- pe… …LL, cECE
- predicti… …er data point
- similar on …AR100 with ResNet28-10 and on ImageNet with ResNet50

Ablation

- weight space analysis
- prediction diversity
- cond. variance of shared activations

Ablation: weight space analysis

- MIMO members converge in non-connected accuracy modes
- naive multihead members converge in the same accuracy mode

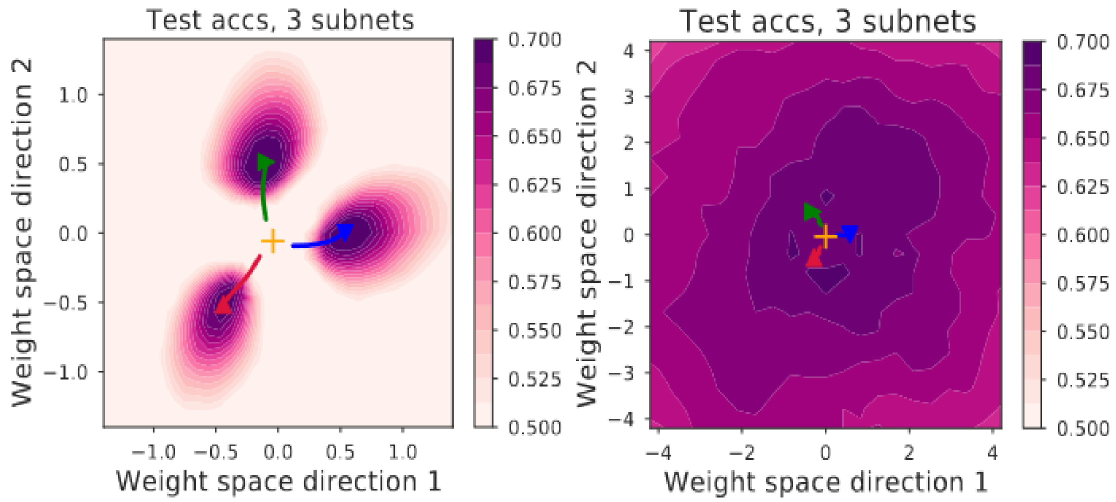

Figure 12: left MIMO, right naive multihead acc, figures made by [1]

- green/red/blue = training trajectories
- don’t display shared weights

Ablation: prediction diversity

- $D_{disagreement} = I(\hat{y}_1 \neq \hat{y}_2)$
- $D_{KL} = KL(p_1,\ p_2)$
- averaged over all head pairs
- performed on the test set

	$D_{\text{disagreement}}$	D_{KL}
Naive multihead	0.000	0.000
TreeNet	0.010	0.010
BatchEnsemble	0.014	0.020
MIMO	0.032	0.086
Deep Ensemble	0.032	0.086

Table 2: average paired head prediction diversity, table made by [1]

Ablation: prediction diversity

- $D_{disagreement} = I(\hat{y}_1 \neq \hat{y}_2)$
- $D_{KL} = KL(p_1, \ p_2)$
- averaged over
- perf

doubt:
unlikely results

	$D_{\text{disagreement}}$	D_{KL}
Naive multihead	0.000	$\boxed{0.000}$
TreeNet	0.010	0.010
BatchEnsemble	0.014	0.020
MIMO	0.032	0.086
Deep Ensemble	0.032	0.086

Table 2: average paired head prediction diversity, table made by [1]

Ablation: cond. variance of shared activations

$$\boldsymbol{var(a|x_2, x_3)} = \mathbb{E}_{\boldsymbol{x_2, x_3}}[\boldsymbol{var_{x_1}(a(x_1, x_2, x_3))}]$$

- toy example with dataset $[x_d, x_e]$ and $M = 3$:

$$\begin{aligned}\boldsymbol{\frac{1}{4}(var\big(a(x_d, x_d, x_d), a(x_e, x_d, x_d)\big)}\\ \boldsymbol{+\, var\big(a(x_d, x_e, x_d), a(x_e, x_e, x_d)\big)}\\ \boldsymbol{+var\big(a(x_d, x_d, x_e), a(x_e, x_d, x_e)\big)}\\ \boldsymbol{+var\big(a(x_d, x_e, x_e), a(x_e, x_e, x_e)\big)}\end{aligned}$$

Ablation: cond. variance of shared activations

$$var(a|x_2, x_3) \; = \mathbb{E}_{x_2,\, x_3}[var_{x_1}(a(x_1, x_2, x_3))]$$

- toy example with dataset $[x_d, x_e]$ and $M = 3$:

$$\begin{aligned}
&\frac{1}{4}\big(var\big(a(x_d, x_d, x_d), a(x_e, x_d, x_d)\big)\\
&+ var\big(a(x_d, x_e, x_d), a(x_e, x_e, x_d)\big)\\
&+ var\big(a(x_d, \boxed{x_d, x_e}), a(x_e, \boxed{x_d, x_e})\big)\\
&+ var\big(a(x_d, x_e, x_e), a(x_e, x_e, x_e)\big)\big)
\end{aligned}$$

Ablation: cond. variance of shared activations

$$var(a|x_2, x_3) \ = \mathbb{E}_{x_2,\, x_3}[var_{x_1}(a(x_1, x_2, x_3))]$$

- toy example with dataset $[x_d, x_e]$ and $M = 3$:

$$\frac{1}{4}(var\big(a(x_d, x_d, x_d), a(x_e, x_d, x_d)\big)$$
$$+ \, var\big(a(x_d, x_e, x_d), a(x_e, x_e, x_d)\big)$$
$$+var\big(a(\boxed{x_d}, x_d, x_e), a(\boxed{x_e}, x_d, x_e)\big)$$
$$+var\big(a(x_d, x_e, x_e), a(x_e, x_e, x_e)\big)$$

Ablation: cond. variance of shared activations

$$\boldsymbol{var(a|x_2, x_3) \ = \mathbb{E}_{x_2,\, x_3}[var_{x_1}(a(x_1, x_2, x_3))]}$$

- toy example with dataset $[x_d, x_e]$ and $M = 3$:

$$\boldsymbol{\frac{1}{4}\Big(var\big(a(x_d, x_d, x_d), a(x_e, x_d, x_d)\big)}$$
$$\boldsymbol{+\, var\big(a(x_d, x_e, x_d), a(x_e, x_e, x_d)\big)}$$
$$\boldsymbol{+\boxed{var\big(a(x_d, x_d, x_e), a(x_e, x_d, x_e)\big)}}$$
$$\boldsymbol{+var\big(a(x_d, x_e, x_e), a(x_e, x_e, x_e)\big)}$$

- (highly) non-zero if changes in x_1 impact a → a belongs to subnet 1
- (close to) zero if changes in x_1 don't impact a → a doesn't belong to subnet 1

Ablation: cond. variance of shared activations

- calculate cond. variance for all subnets
- many a_i belong to only 1 subnet

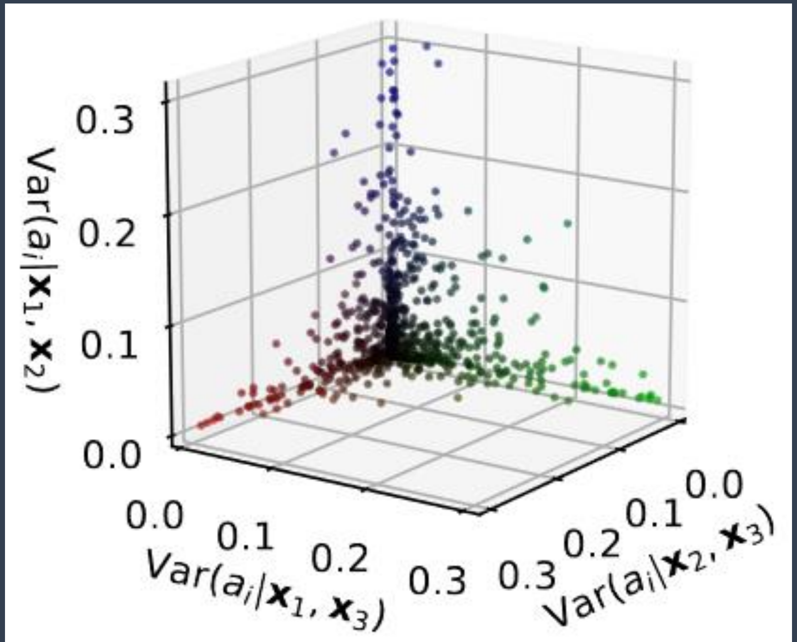

Figure 13: cond. variances of hidden activations, figure made by [1]

- red:
- $\boldsymbol{var(a|x_2, x_3)}$ non-zero
- $\boldsymbol{var(a|x_1, x_2)}$ & $\boldsymbol{var(a|x_1, x_3)}$ zero

  → red a_i belong to subnet 1 only

Hyperparameters

- M
- input repetition
- batch repetition

Hyperparameters: M

- too small: ensemble not fully leveraged (worse generalization & calibration)
- too big: can exceed models capacity (members are too weak)
- more L1/2 penalty → smaller $M_{optimal}$ (shows that MIMO exploits capacity)

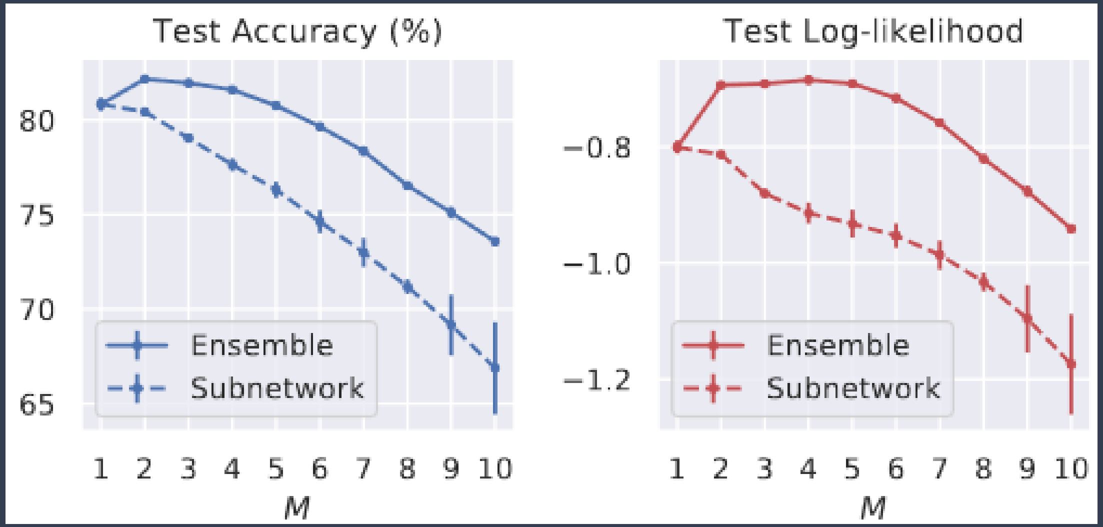

Figure 14: varying M on CIFAR100/ResNet28-10, figure made by [1]

- log-likelihood peaks later (benefits more from larger ensemble)
- similar results on CIFAR10

Hyperparameters: input repetition

- feed same input into multiple subnets in one forward pass with certain prob. (training mode)
- allows independent subnetworks for models with low capacity
- determines the independence of the subnetworks

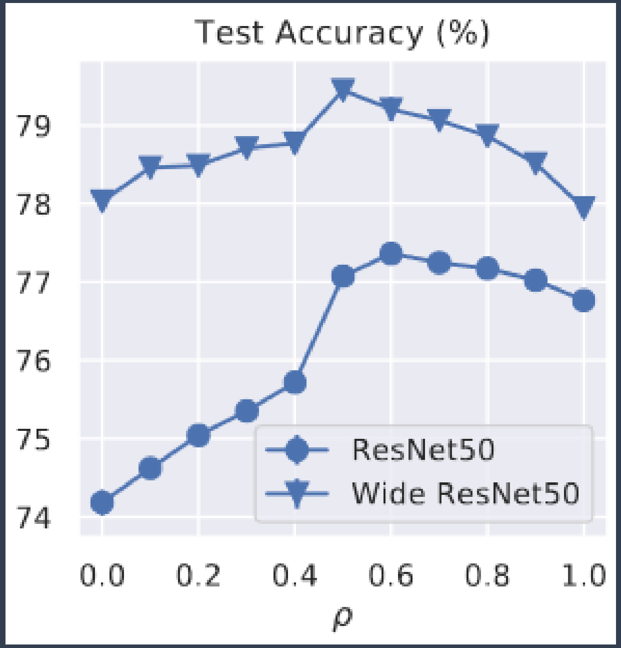

Figure 15: input repetition on ImageNet, figure made by [1]

Hyperparameters: batch repetition

- $batch_{br} = concat(N\ times\ batch)$

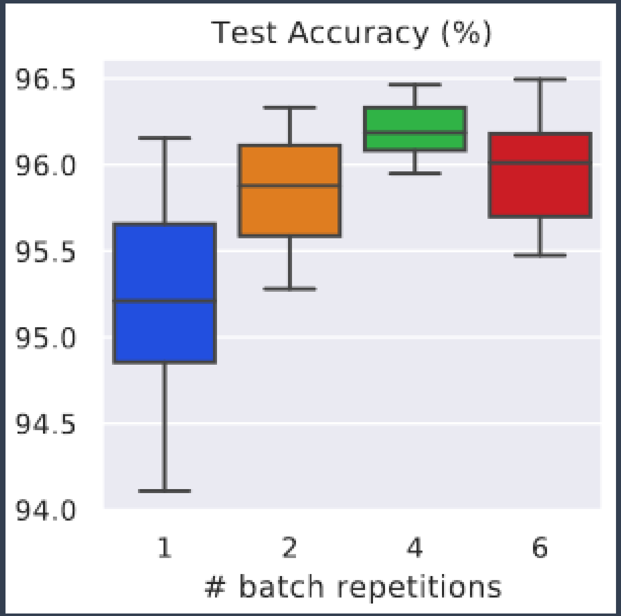

Figure 16: batch repetition (12 runs), figure made by [1]

- 12 runs varying in M, lr & batch size

Hyperparameters: batch repetition

- $batch_{br} = concat(N\ times\ batch)$

probably works when repetitions are shuffled (x_i in $m > 1$ subnets) (not mentioned)

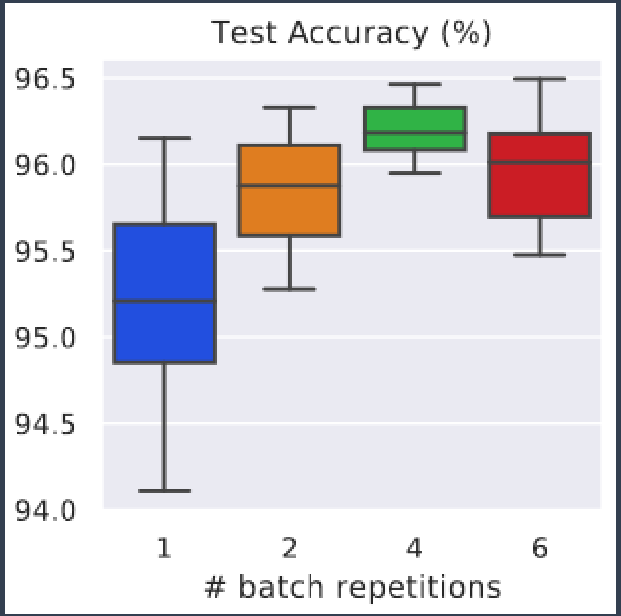

Figure 16: batch repetition (12 runs), figure made by [1]

- 12 runs varying in M, lr & batch size

Summary

- ensemble with independent members
- few space & time complexity increase
- increase calibration & generalization
- can leverage the capacity of a network

References

[1] Marton Havasi, Rudolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew M. Dai, Dustin Tran (2021):
„Training independent subnetworks for robust prediction“,
International Conference on Learning Representations,
digital, 03.05.2021 – 07.05.2021.

[2] Balaji Lakshminarayanan, Alexander Pritzel, Charles Blundell (2017):
„Simple and scalable predictive uncertainty estimation using deep ensembles“,
Advances in neural information processing systems, Vol. *30, p. 6402 – 6413,*
Morgan Kaufmann Publishers Inc..

[3] Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David Crandall, Dhruv Batra (2015):
„Why M heads are better than one: Training a diverse ensemble of deep networks“,
arXiv:1511.06314.

[4] Yeming Wen, Dustin Tran, Jimmy Ba (2020):
"BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning",
International Conference on Learning Representations,
Digital, 26.04.2020 – 01.05.2020.

[5] Gal Yarin, Zoubin Ghahramani (2016):
"Dropout as a bayesian approximation: Representing model uncertainty in deep learning",
international conference on machine learning,
USA, New York City, 19.06.2016 – 24.06.2016.