DARTMOUTH
Department of Computer Science

ICML
International Conference
On Machine Learning

Slot Machines: Discovering Winning Combinations of Random Weights in Neural Networks

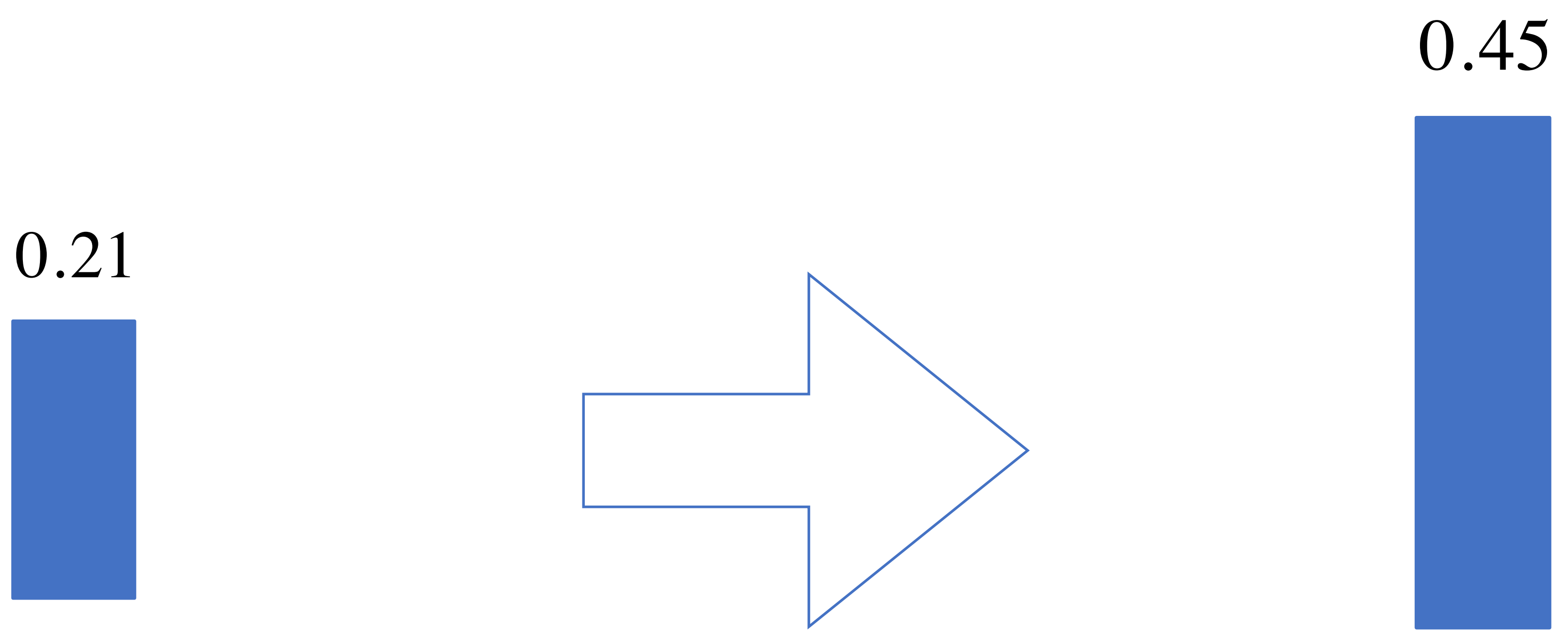
Maxwell Aladago, Lorenzo Torresani

DARTMOUTH
Department of Computer Science

ICML
International Conference On Machine Learning

Motivation

In traditional neural network optimization, each connection is assigned 1 continuous weight which is updated in each iteration



$w_{i,j}^{(t)}$ $\quad$ $w_{i,j}^{(t+1)} \leftarrow w_{i,j}^{(t)} - \alpha \frac{\partial L}{\partial w_{i,j}}$ $\quad$ $w_{i,j}^{(t+1)}$

Can selection from **a fixed set of random weights** be competitive with traditional continuous weight optimization?

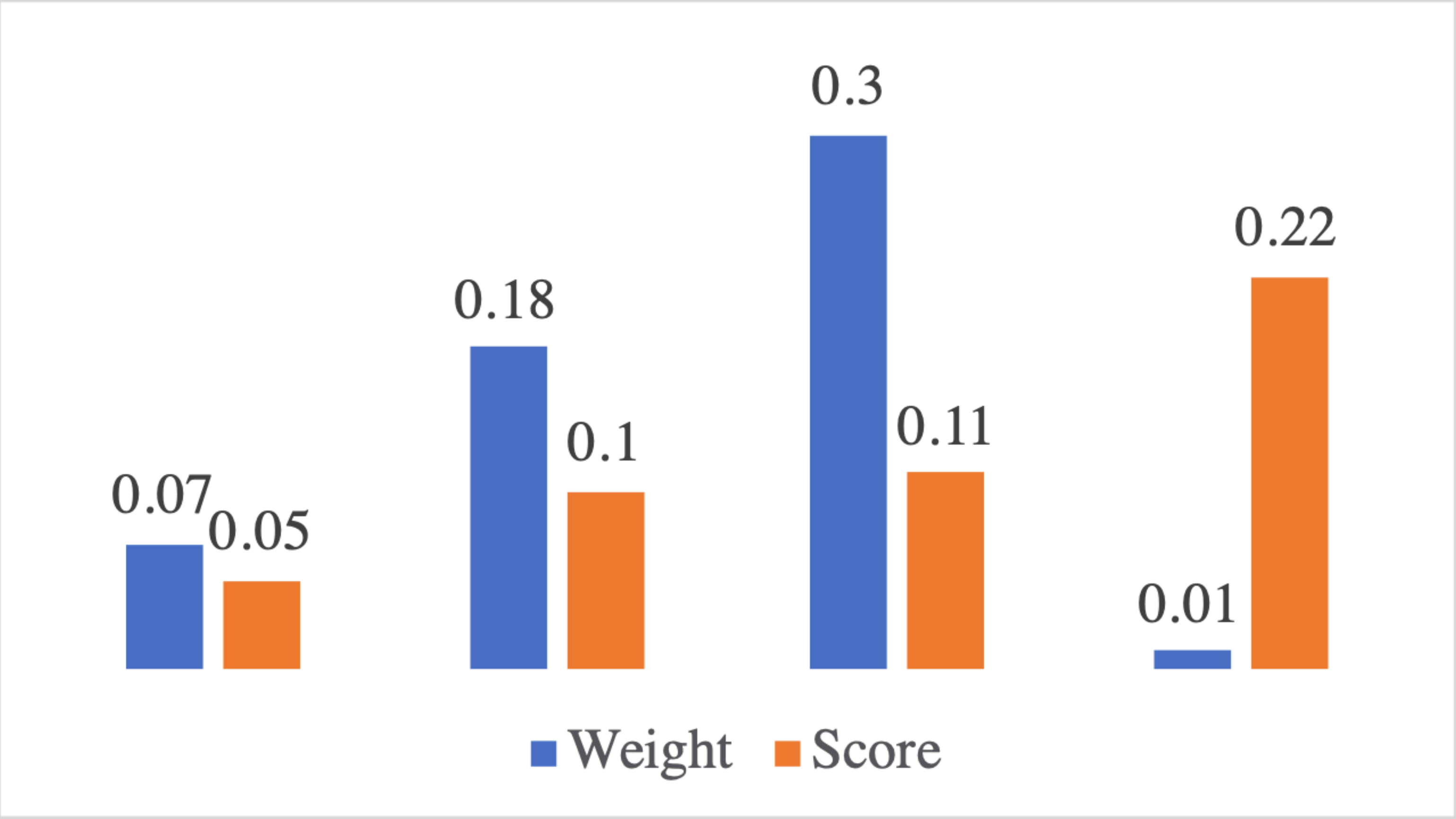
Introducing Slot Machines

We present a family of neural networks called “slot machines” where each reel (connection) contains **a fixed set of symbols (random values),** and a backpropagation algorithm that “spins” the reels to seek “winning” combinations, i.e., selections of values that minimize the given loss.

DARTMOUTH
Department of Computer Science

Slot Machines

Key Idea: select 1 weight from K fixed random values per connection based on associated scores



Connection (i, j) at iteration t

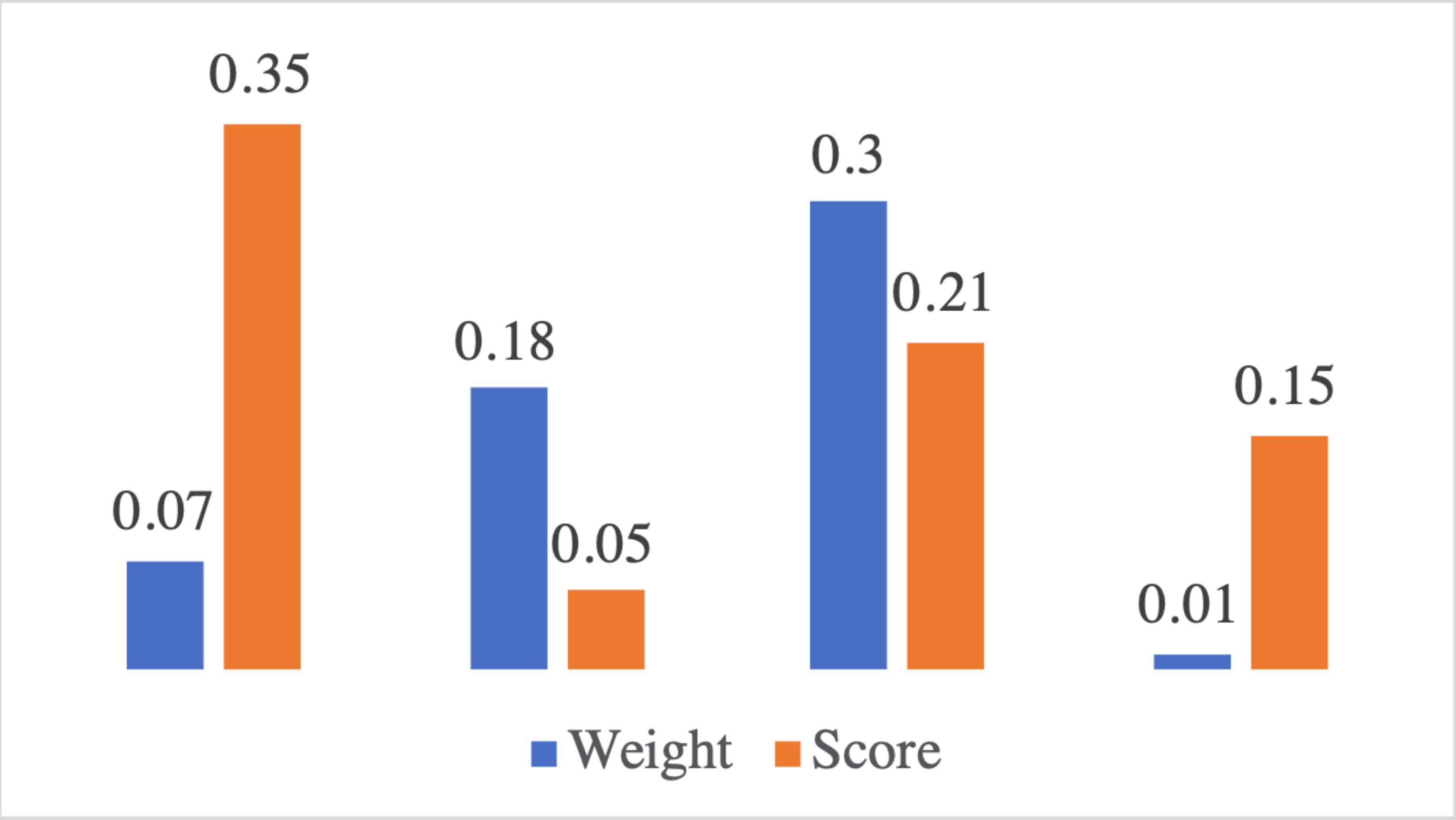
Forward Pass

Select one of the K weights according to the scores

$w_{i,j}$

Loss L

Backward Pass



Connection (i, j) at iteration $t+1$

$$s_{i,j,k}^{(t+1)} \leftarrow s_{i,j,k}^{(t)} - \alpha \frac{\partial L}{\partial a(x)_i^{(\ell)}} h(x)_j^{(\ell-1)} W_{i,j,k}$$

Approach

Using multiple weight values per connection increases the chances of finding a combination of weights that is competitive

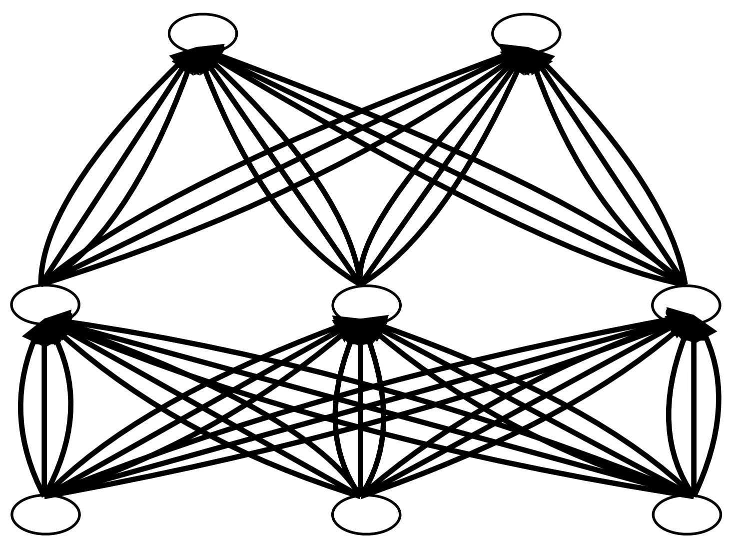
Initialization:

- ➡ Assign K random values to each connection (i, j)
- ➡ Initialize K "quality scores", one for each weight

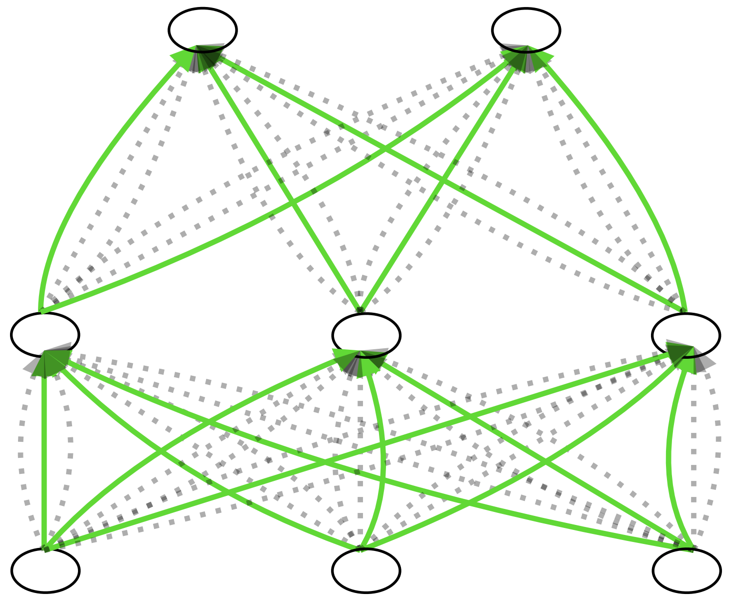
Optimization:

- ➡ In the forward pass, select 1 of the K weights using the scores.
- ➡ In the backward pass, update all the scores using a straight-through gradient estimator (Bengio et al., 2013)

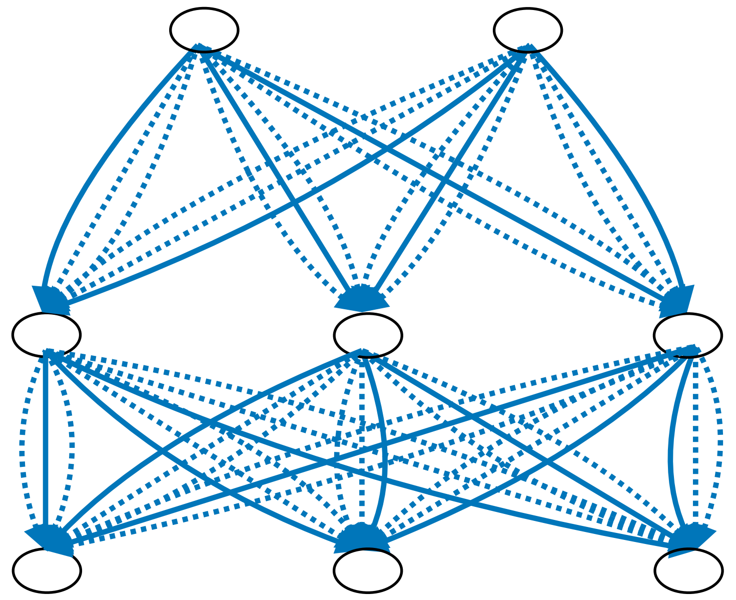
For each edge (i, j), assign K fixed random weights and K corresponding scores.



Forward: use only one of the K weight options for each edge base on the learned scores.



Backward: update all scores including those that were not used in the forward pass.



Selection Methods

Using multiple weight values per connection increases the chances of finding a combination of weights that is competitive

Initialization:

- ➡ Assign K random values to each connection (i, j)
- ➡ Initialize K "quality scores", one for each weight

Optimization:

- ➡ In the forward pass, select 1 of the K weights using the scores.
- ➡ In the backward pass, update all the scores using a straight-through gradient estimator (Bengio et al., 2013)

Selection Methods: $\rho \in \{1, \cdots, K\}$: selected weight index

1. **Greedy Selection (GS)**: $\rho = \arg\max_k \left(\left\{ s_{ij1}, \cdots, s_{ijK} \right\} \right)$
2. **Probabilistic Selection (PS)**: $\rho \sim \text{Mult}\left(\frac{e^{s_{ij1}}}{\sum_{k'=1}^{K} e^{s_{ijk'}}}, \cdots, \frac{e^{s_{ijK}}}{\sum_{k'=1}^{K} e^{s_{ijk'}}} \right)$

Bengio, Y., Le´onard, N., and Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013

Experimental Setup

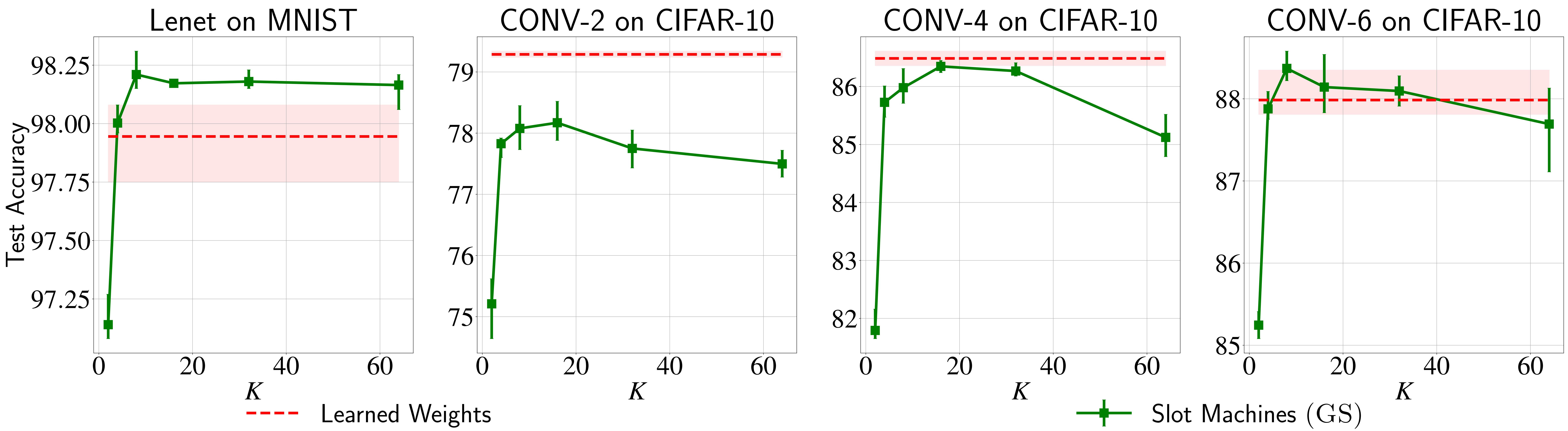
We evaluate Slot Machines on MNIST and CIFAR-10 using five different networks

Network	Lenet	CONV-2	CONV-4	CONV-6	VGG-19
Convolutional Layers		64, 64, pool	64, 64, pool 128, 128, pool	64, 64, pool 128, 128, pool 256, 256, pool	2x64, pool 2x128, pool 2x256, pool 4x512, pool 4x512, avg-pool
Fully-connected Layers	300, 100, 10	256, 256, 10	256, 256, 10	256, 256, 10	10
Epochs: Slot Machines	200	200	200	200	220
Epochs: Learned Weights	200	200	330	330	320
Dataset	MNIST	CIFAR-10	CIFAR-10	CIFAR-10	CIFAR-10

Network Architectures: ‘pool’ indicates max-pooling. All convolutions use 3×3 filters

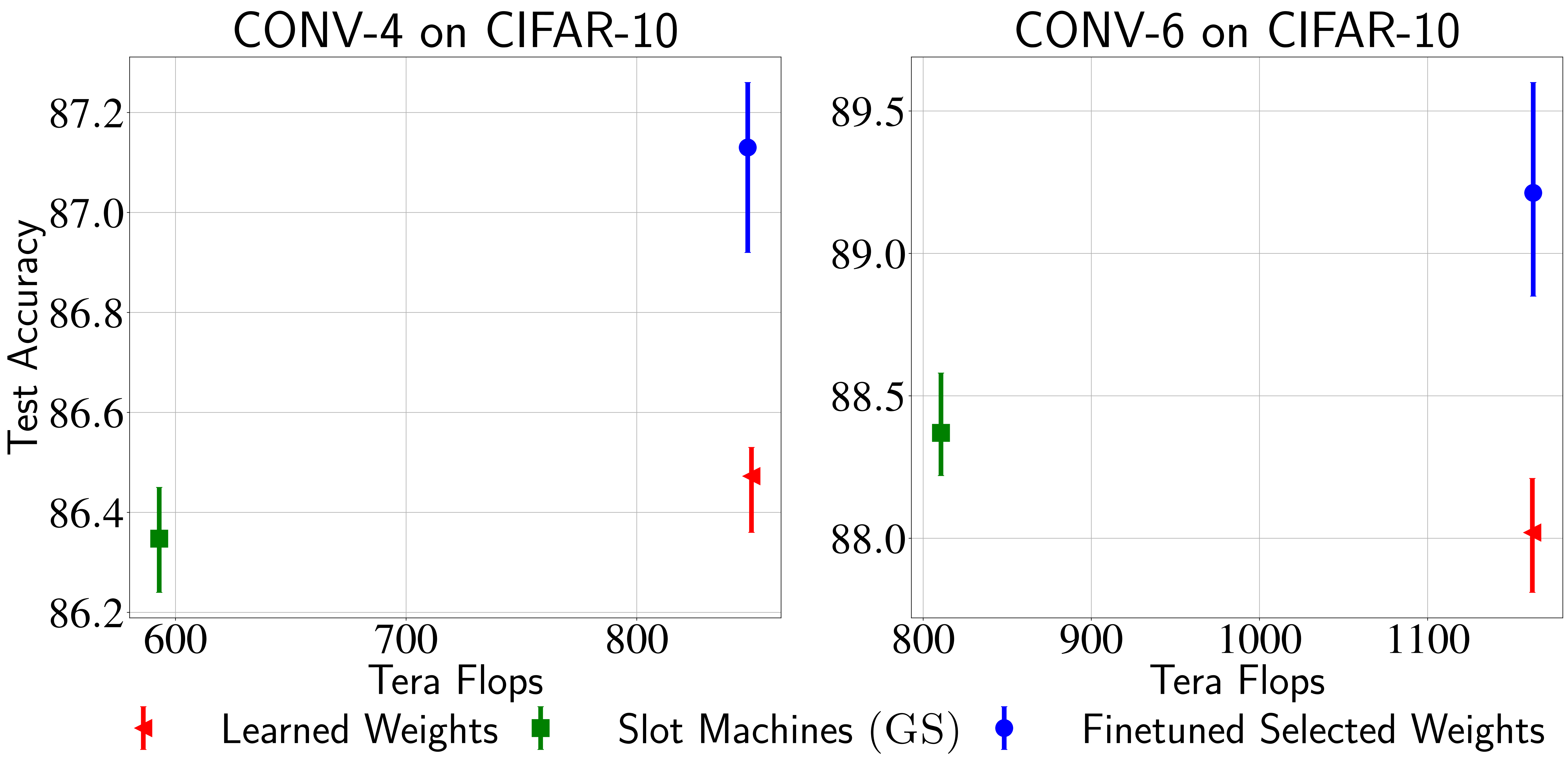
Comparison between Slot Machines and Traditional Neural Networks

Despite containing only random values, the performance of Slot Machines approaches that of traditionally-trained networks if the number of random weight options (K) is large enough, e.g., $K = 8$.



Finetuning Slot Machines

For the same total training budget, finetuned Slot Machines produce higher accuracy compared to the same models trained from scratch.



Comparisons with Related Works

Pruning randomly initialized networks either through

- greedy selection (Ramanujan et al., 2020), or
- probabilistic selection (Zhou et al., 2019).

Slot Machines do not employ any pruning and have multiple options per connection

Method	Lenet	CONV-2	CONV-4	CONV-6
Ramanujan et al. (2020)	-	77.7	85.8	88.1
Supermask (Zhou et al., 2019)	98.0	66.0	72.5	76.5
Slot Machines (GS)	98.2	78.2	86.3	88.4
Slot Machines (PS)	98.0	71.7	80.2	81.7

Zhou, H., Lan, J., Liu, R., and Yosinski, J. De- constructing lottery tickets: Zeros, signs, and the supermask. In Wallach, H., Larochelle, H., Beygelz- imer, A., d'Alche´-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 3597–3607.

Ramanujan, V., Wortsman, M., Kembhavi, A., Farhadi, A., and Rastegari, M. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF Conference on Computer Vi- sion and Pattern Recognition (CVPR)*, June 2020.

Conclusion & Future Work

- This work demonstrates that neural networks with random weights perform competitively, given multiple weight options per connection and a good selection protocol
- Simply selecting the weight with the greatest score is remarkably effective at obtaining competitive weight configurations from Slot Machines
- Finetuning selected configurations from Slot Machines often produces accuracy gains over training the network from scratch, at comparable computational cost
- Future work will analyze the properties that differentiate the selected weights from those that are not selected. Knowing such properties can motivate effective neural network initialization methods