Meta-Learning Bidirectional Update Rules

Mark Sandler, Max Vladymyrov, Andrey Zhmoginov, Nolan Miller, Andrew Jackson, Tom Madams, Blaise Agüera y Arcas

Goal

Meta-learn synapse update rules with very mild assumptions on the inner-loop (no loss functions, no gradients) that learns faster than traditional methods.

Motivation

Meta-Learning Bidirectional Update Rules
Mark Sandler, Max Vladymyrov, Andrey Zhmoginov, Nolan Miller, Andrew Jackson, Tom Madams, Blaise Agüera y Arcas

SGD is a special case of two-state neurons
Backpropagation can be equivalently reformulated with generalized two-state neurons, where \( \gamma \) is a layer and \( \gamma \in \{0, 1\} \) is a state.

Bidirectional Learning Update Rules (BLUR): 
- Synapse updated rules are parametrized and meta-learned via a low-dimensional genome matrix.
- No predefined per-iteration loss function, no explicit gradients.
- Keep bidirectionality of the updates:
  - Input is passed at the forward pass,
  - Labels are passed at the backward pass.
- Metatrain to a given iteration (unroll).

SGD optimization via Backpropagation:
- Uses predefined loss function computed at every iteration.
- The loss is minimized via gradient descent (steepest direction of the current loss).
- Optimization can use previous iterations (e.g. momentum), but (mostly) can’t see forward.
- Optimization procedure is independent from the dataset.

SGD w/ different parameters vs BLUR
Genome learns faster than SGD with any learning rate/momentum.

Generalization of a genome
- Trained on 10x10 MNIST using 2-layer 4-state architecture. Validated on 28x28 digits.

Bidirectional Learning Update Rules (BLUR)

Bidirectional Learning Update Rules (BLUR): 
- Synapse updated rules are parametrized and meta-learned via a low-dimensional genome matrix.
- No predefined per-iteration loss function, no explicit gradients.
- Keep bidirectionality of the updates:
  - Input is passed at the forward pass,
  - Labels are passed at the backward pass.
- Metatrain to a given iteration (unroll).

SGD optimization via Backpropagation:
- Uses predefined loss function computed at every iteration.
- The loss is minimized via gradient descent (steepest direction of the current loss).
- Optimization can use previous iterations (e.g. momentum), but (mostly) can’t see forward.
- Optimization procedure is independent from the dataset.

SGD w/ different parameters vs BLUR
Genome learns faster than SGD with any learning rate/momentum.

Generalization of a genome
- Trained on 10x10 MNIST using 2-layer 4-state architecture. Validated on 28x28 digits.

Bidirectional Learning Update Rules (BLUR)

Bidirectional Learning Update Rules (BLUR): 
- Synapse updated rules are parametrized and meta-learned via a low-dimensional genome matrix.
- No predefined per-iteration loss function, no explicit gradients.
- Keep bidirectionality of the updates:
  - Input is passed at the forward pass,
  - Labels are passed at the backward pass.
- Metatrain to a given iteration (unroll).

SGD optimization via Backpropagation:
- Uses predefined loss function computed at every iteration.
- The loss is minimized via gradient descent (steepest direction of the current loss).
- Optimization can use previous iterations (e.g. momentum), but (mostly) can’t see forward.
- Optimization procedure is independent from the dataset.

SGD w/ different parameters vs BLUR
Genome learns faster than SGD with any learning rate/momentum.

Generalization of a genome
- Trained on 10x10 MNIST using 2-layer 4-state architecture. Validated on 28x28 digits.

Bidirectional Learning Update Rules (BLUR)

Bidirectional Learning Update Rules (BLUR): 
- Synapse updated rules are parametrized and meta-learned via a low-dimensional genome matrix.
- No predefined per-iteration loss function, no explicit gradients.
- Keep bidirectionality of the updates:
  - Input is passed at the forward pass,
  - Labels are passed at the backward pass.
- Metatrain to a given iteration (unroll).

SGD optimization via Backpropagation:
- Uses predefined loss function computed at every iteration.
- The loss is minimized via gradient descent (steepest direction of the current loss).
- Optimization can use previous iterations (e.g. momentum), but (mostly) can’t see forward.
- Optimization procedure is independent from the dataset.

SGD w/ different parameters vs BLUR
Genome learns faster than SGD with any learning rate/momentum.

Generalization of a genome
- Trained on 10x10 MNIST using 2-layer 4-state architecture. Validated on 28x28 digits.

Bidirectional Learning Update Rules (BLUR)

Bidirectional Learning Update Rules (BLUR): 
- Synapse updated rules are parametrized and meta-learned via a low-dimensional genome matrix.
- No predefined per-iteration loss function, no explicit gradients.
- Keep bidirectionality of the updates:
  - Input is passed at the forward pass,
  - Labels are passed at the backward pass.
- Metatrain to a given iteration (unroll).

SGD optimization via Backpropagation:
- Uses predefined loss function computed at every iteration.
- The loss is minimized via gradient descent (steepest direction of the current loss).
- Optimization can use previous iterations (e.g. momentum), but (mostly) can’t see forward.
- Optimization procedure is independent from the dataset.

SGD w/ different parameters vs BLUR
Genome learns faster than SGD with any learning rate/momentum.

Generalization of a genome
- Trained on 10x10 MNIST using 2-layer 4-state architecture. Validated on 28x28 digits.