A projection-based framework for gradient-free and parallel learning.

From loss minimization to feasibility seeking.

Andreas Bergmeister [1] Manish Krishan Lal [1] Stefanie Jegelka [1,2] Suvrit Sra [1,2]

[1] TU Munich, MCML · [2] MIT

AISTATS 2026 GITHUB.COM/ANDREASBERGMEISTER/PJAX

THE STATUS QUO

Learning is framed as loss minimization.

FORMULATION

Empirical risk minimization

$\min_\theta \sum_i \ell(f(x_i; \theta), y_i)$

GRADIENTS

Backpropagation

SOFTWARE

PyTorch · TensorFlow · JAX

OPTIMIZER

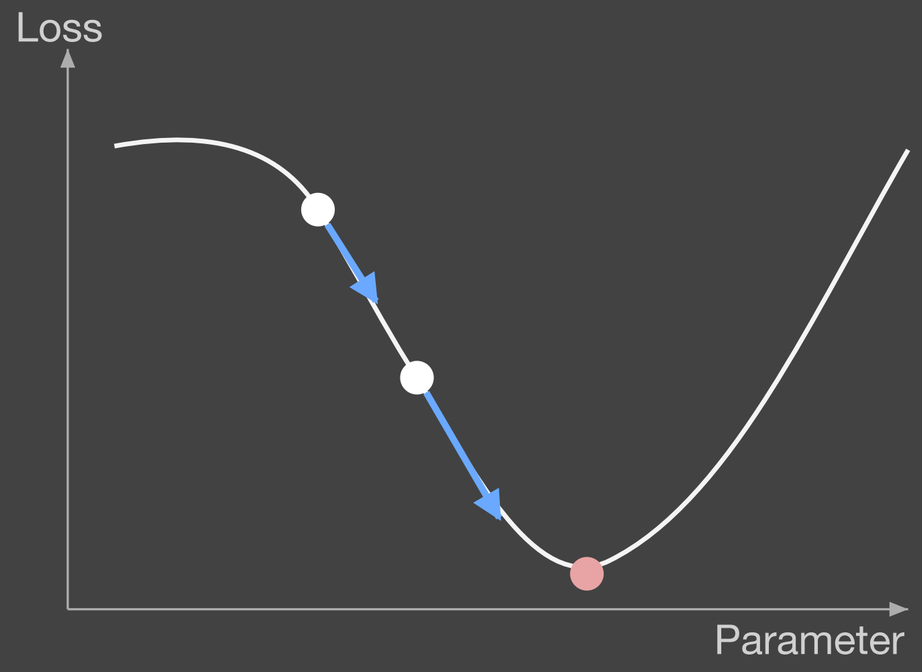
SGD · Adam · etc.

A NEW FORMULATION

From loss minimization to feasibility seeking.

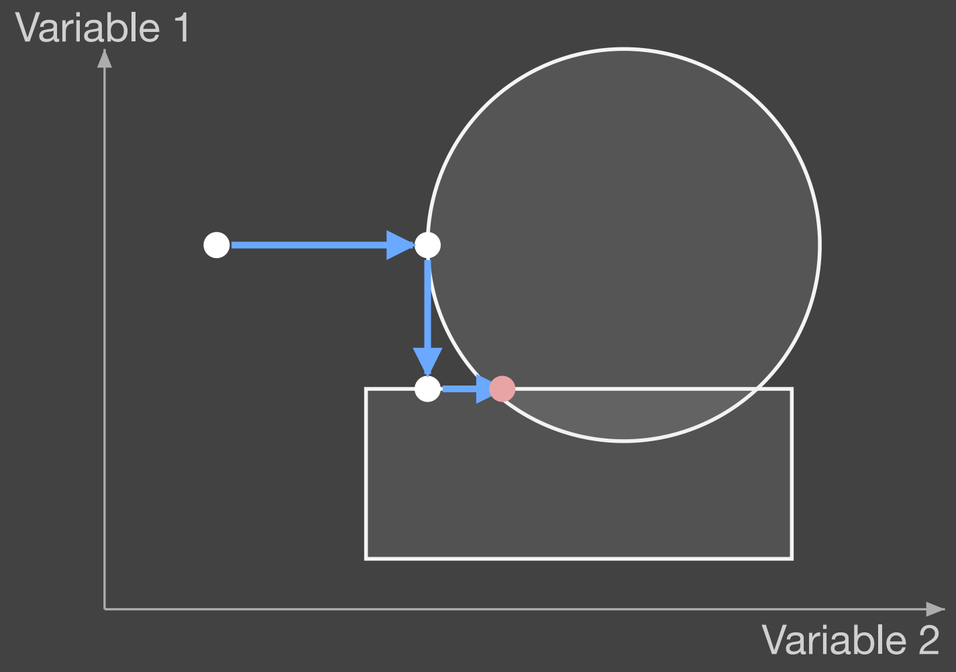
GRADIENT-BASED MINIMIZATION

Descend a scalar loss surface in parameter space.



PROJECTION-BASED FEASIBILITY

Find a state (parameters + activations) in the intersection of constraints.

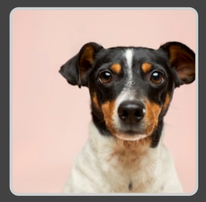


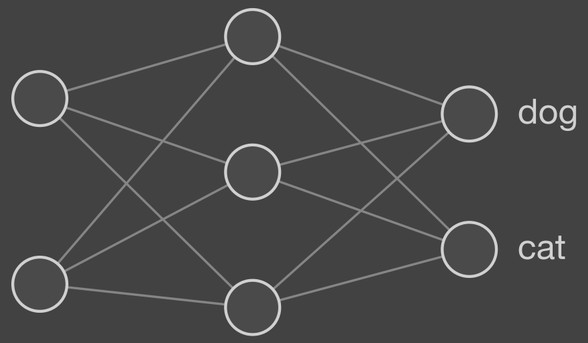
FROM DATA AND ARCHITECTURE TO FEASIBILITY

Data and architecture become constraints.

DATA CONSTRAINT

Each example constrains the network output.



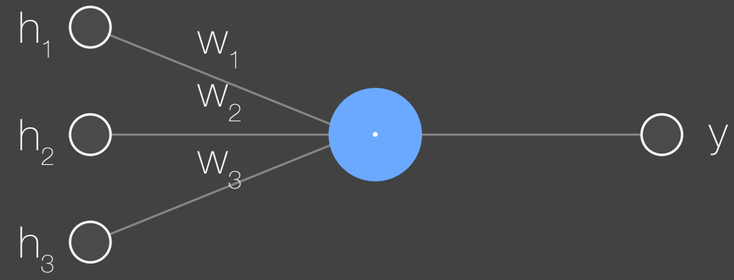


OUTPUT CONSTRAINT

dog > cat

ARCHITECTURE CONSTRAINT

Each primitive function constrains its inputs and output.



PRIMITIVE CONSTRAINT

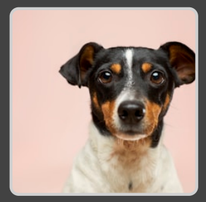
$\langle h, w \rangle = y$

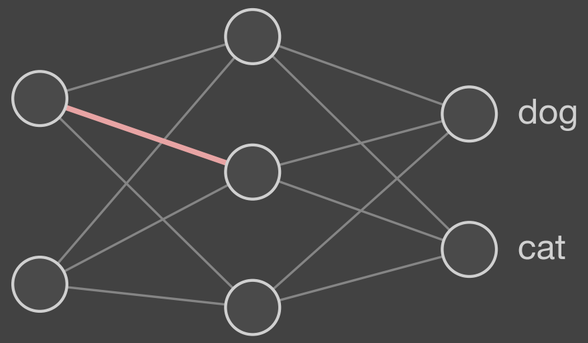
FROM DATA AND ARCHITECTURE TO FEASIBILITY

Data and architecture become constraints.

DATA CONSTRAINT

Each example constrains the network output.



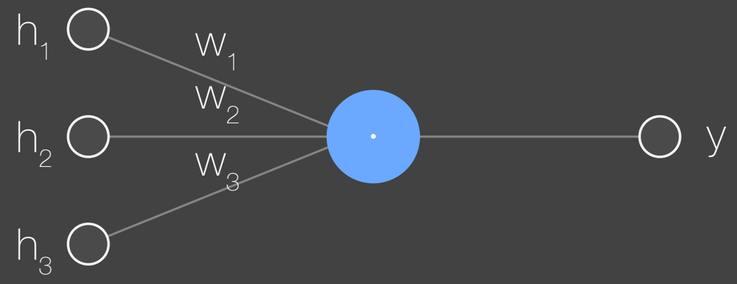


OUTPUT CONSTRAINT

dog > cat

ARCHITECTURE CONSTRAINT

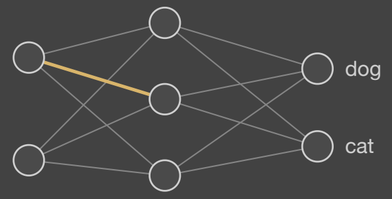
Each primitive function constrains its inputs and output.



PRIMITIVE CONSTRAINT

$\langle h, w \rangle = y$

ANOTHER SAMPLE



PARAMETER CONSENSUS

$W^{(1)} = W^{(2)}$

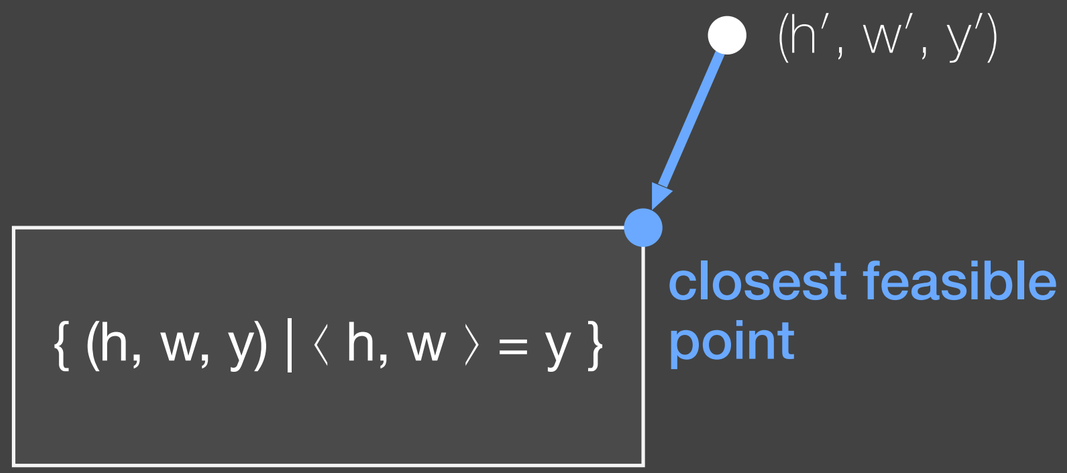
copies of shared weights must agree

PROJECTION METHODS

Projection methods iterate toward feasibility.

SINGLE PROJECTION

Move an infeasible state to the nearest point on one constraint set.

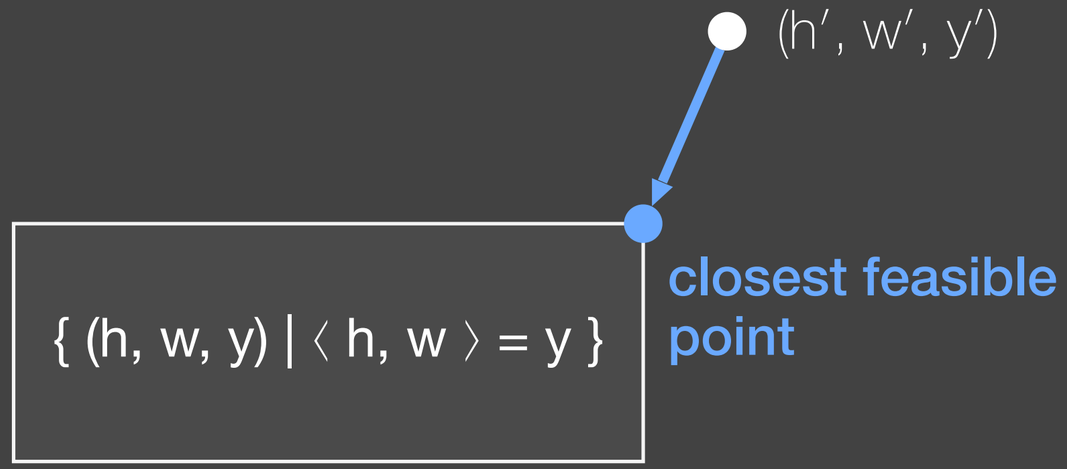


PROJECTION METHODS

Projection methods iterate toward feasibility.

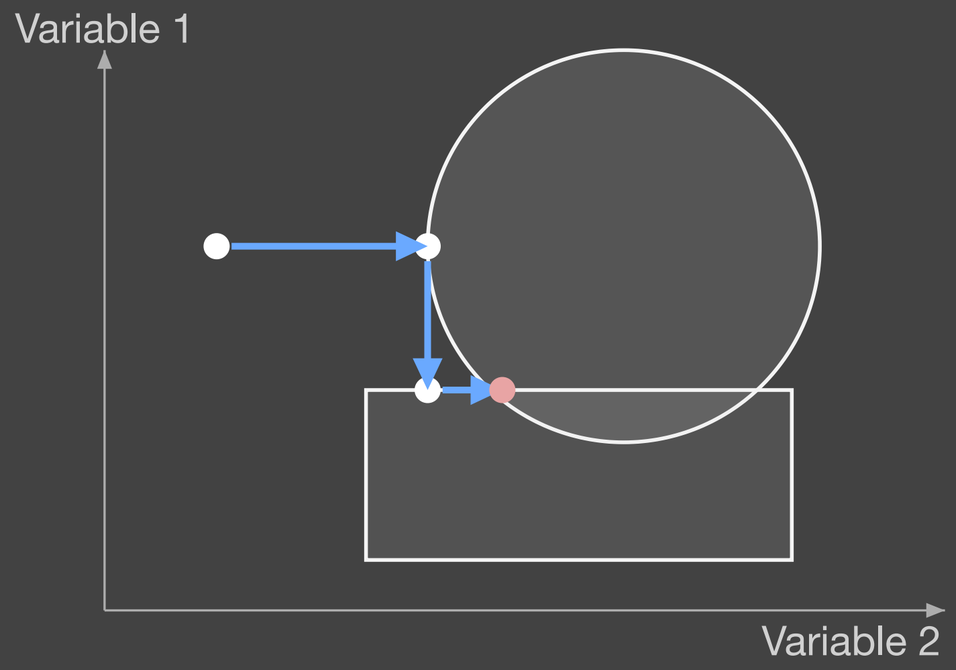
SINGLE PROJECTION

Move an infeasible state to the nearest point on one constraint set.



PROJECTION METHOD

Repeat projections across constraint sets to reach their intersection.

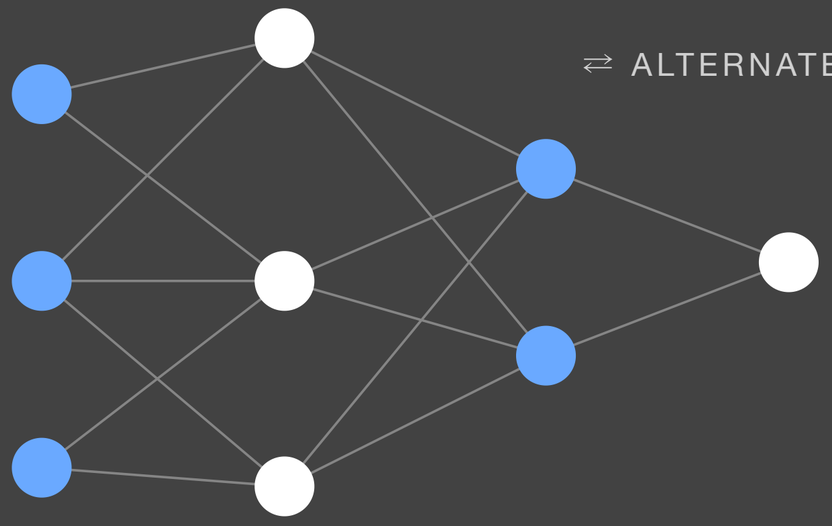


PARALLEL PROJECTION STEPS

Local projections enable parallel updates.

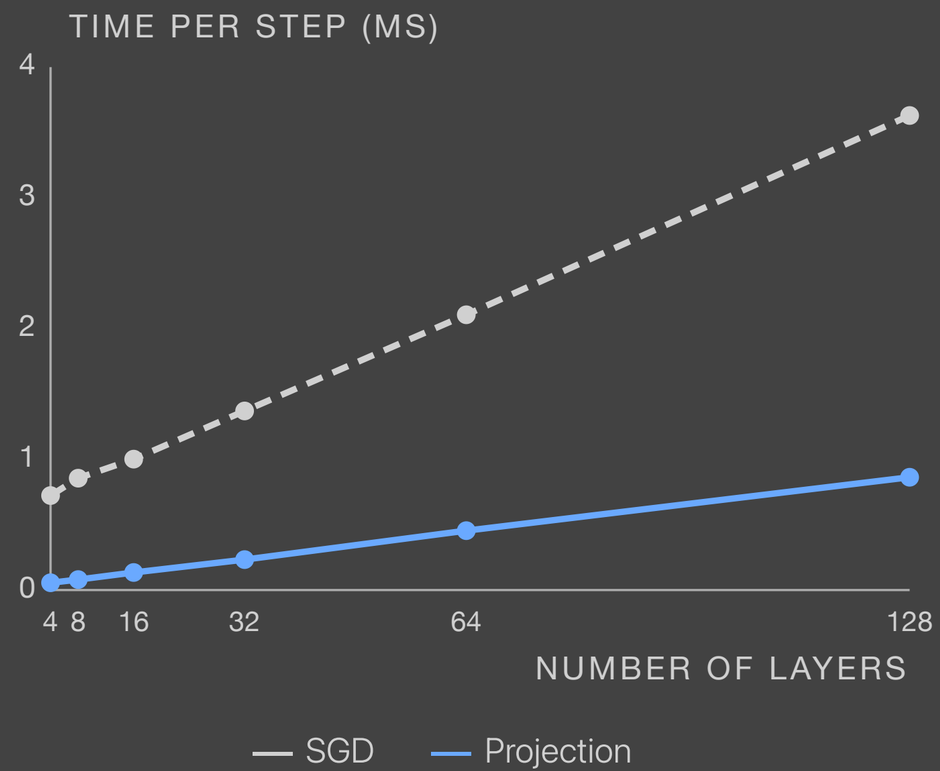
GRAPH BIPARTITION

Project all nodes in one partition in parallel, then alternate.



SCALABILITY

Parallel projections scale more gently with depth.



THE SOFTWARE LAYER

PJAX is the projection analogue of autodiff.

TRAINING.PY

```python
from pjax import nn, optim

inputs, targets = batch
model  = MLP(784, 256, 10)
params = model.init(key)
solver = optim.DouglasRachford()

def loss_fn(params):
    logits = model.apply(params, inputs)
    return nn.cross_entropy(logits, targets)

params, loss = solver.update(loss_fn, params)
```

01 **USER**
Write a forward computation in a familiar API.

↓

02 **TRACE**
Build the computation graph for a batch.

↓

03 **ATTACH**
Attach projection operators to primitive constraints.

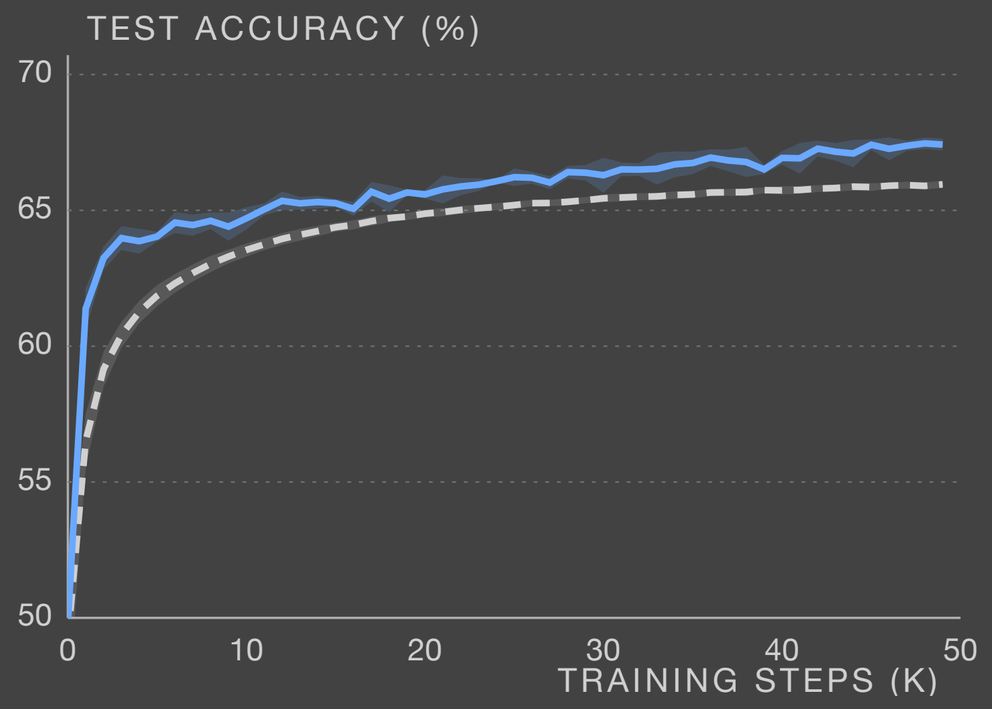
↓

04 **SOLVE**
Iterate projections and extract parameters.

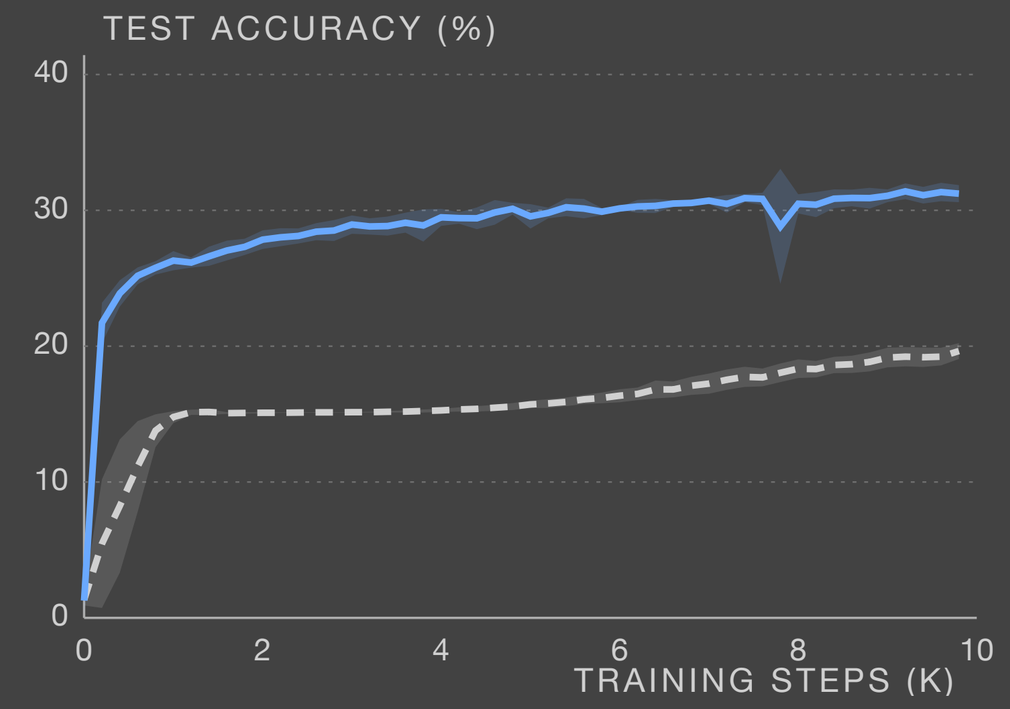
EMPIRICAL VALIDATION

Projection-based training can improve over SGD.

HIGGS · MLP — 1 × 256 NEURONS



Shakespeare · RNN — 1 × 256 NEURONS



SGD — Projection

GITHUB.COM/ANDREASBERGMEISTER/PJAX