# Mode Connectivity in Neural Networks

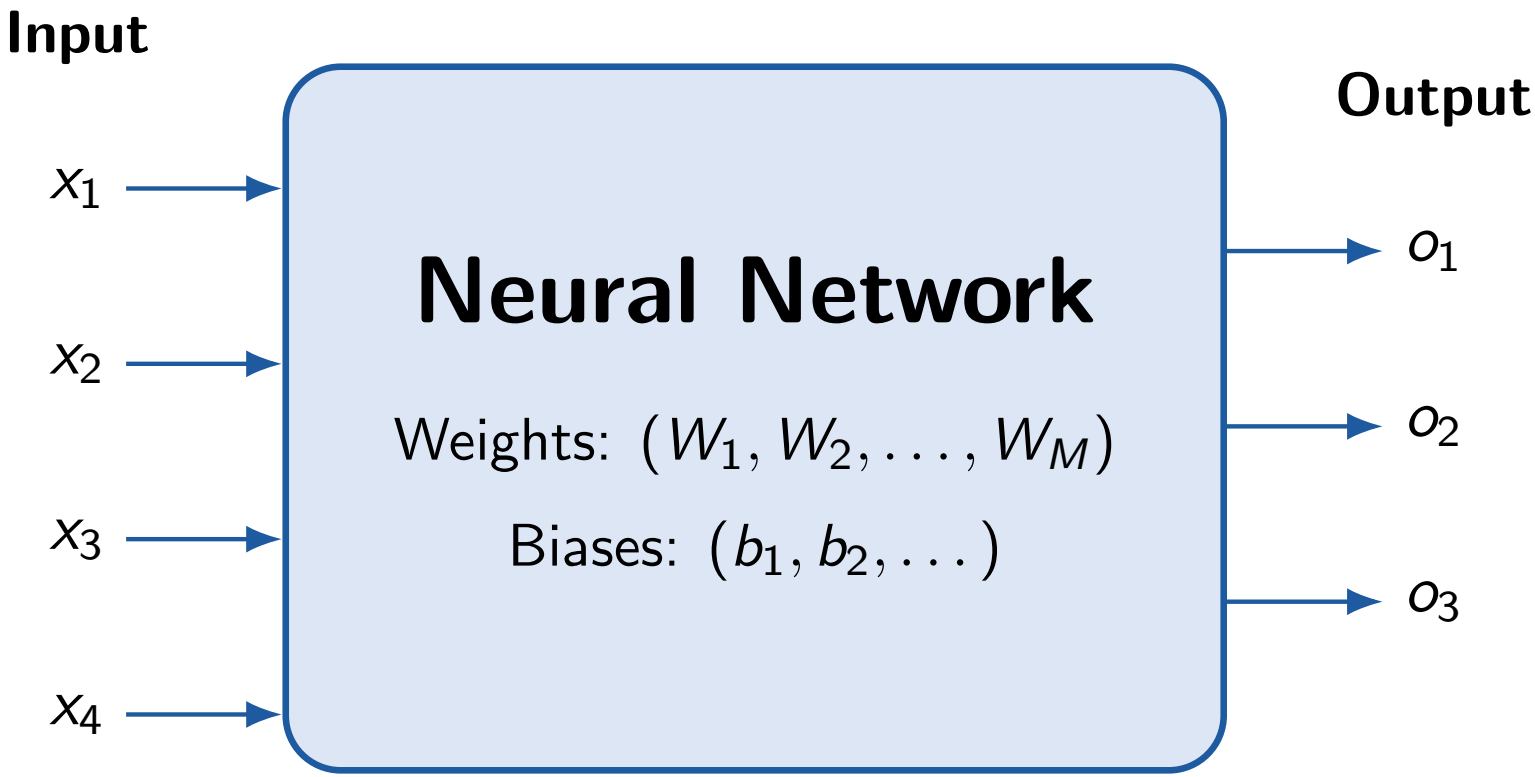
## Literature Review

Dr. Tuhin Subhra Mukherjee

Based on the works of
**Garipov et al. (arXiv:1802.10026v4)** and **Draxler et al. (arXiv:1803.00885v5)**

February 26, 2026

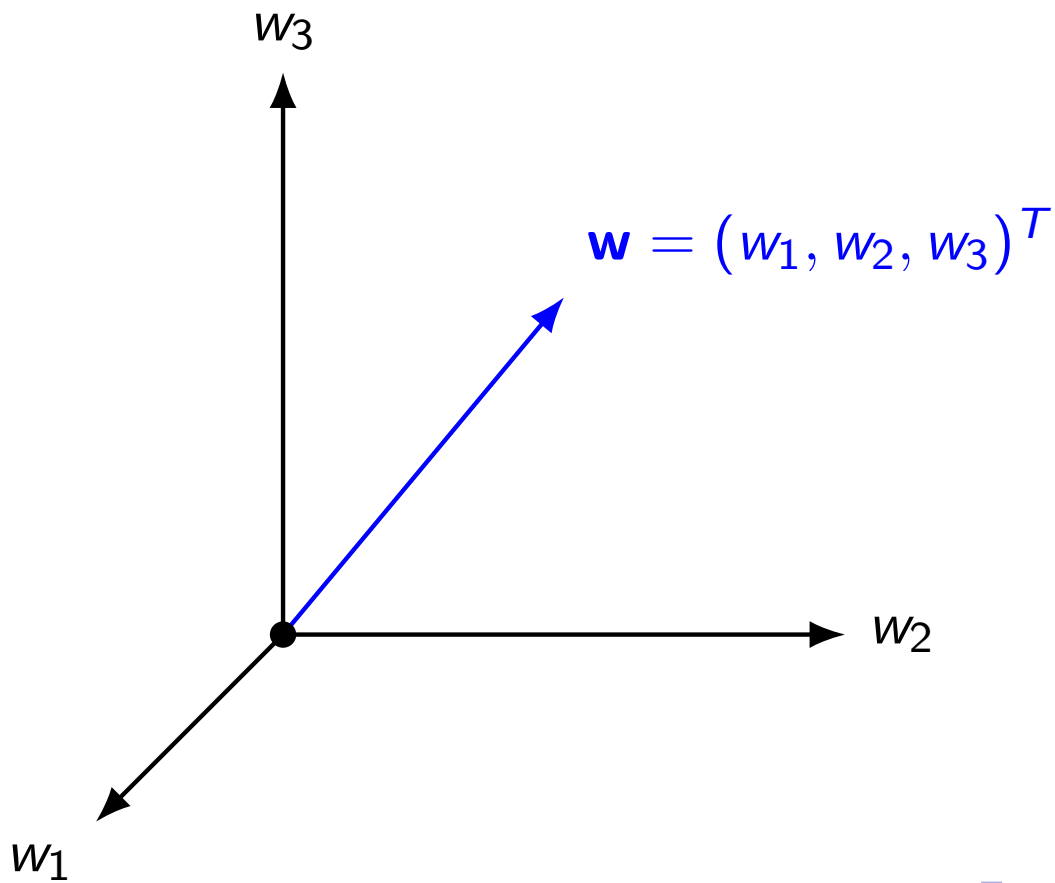
# Cartoon of Neural Network




- Collect all the weights in a vector $\vec{W} = (w_1, w_2, \cdots w_M)^T$.
- Collect the input variables in a vector $\vec{X} = (x_1, x_2 \cdots x_P)^T$
- Collect the output variables in vector $\vec{O} = (o_1, o_2, \cdots o_Q)^T$.
- We have $\vec{O} = \vec{O}(\vec{X}, \vec{W}.)$

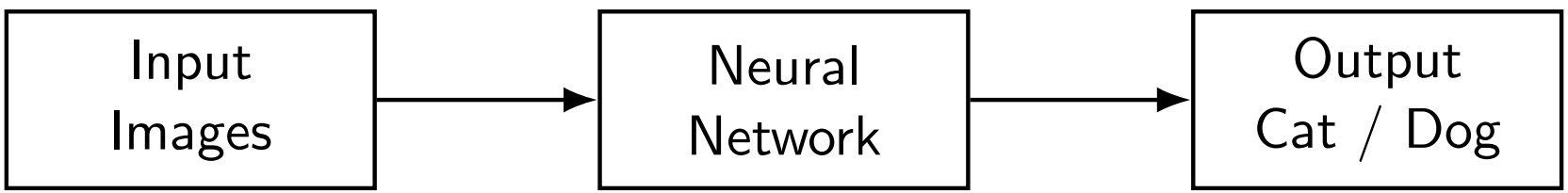
# Weight Space

- Suppose there are $M$ weights. Then the space of weights is an $M$-dimensional vector space.
- Each vector (point) in the weight space gives a choice of parameters.



# What is a Neural Network Good For?

- **Example Task (Binary Classification):**
  Suppose we feed a collection of *cat* and *dog* images as input. We want the neural network to produce outputs that correctly identify whether an image is a cat or a dog.




  - This is accomplished by **"training"** the network.

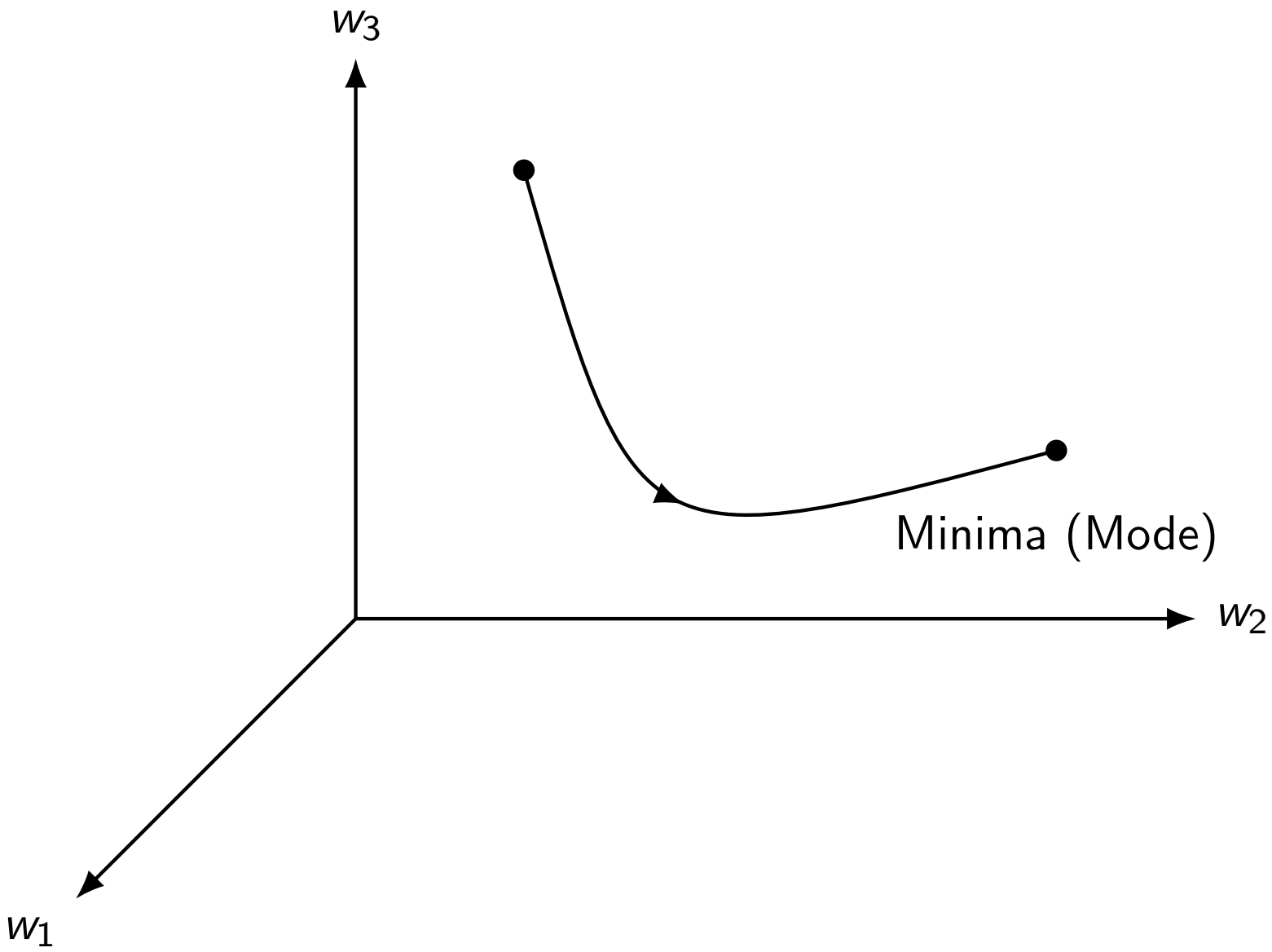
# Training procedure

- Suppose we are given $N$ data points $\{\vec{X_1}, \vec{X_2} \cdots \vec{X_n}\}$ with the corresponding true labels $\{\vec{T_1}, \vec{T_2}, \cdots \vec{T_N}\}$.
- We define a loss function $L$ which is a function of weight vector $\vec{W}$, input data $\{\vec{X_i}\}_{i=1}^{N}$, and the true labels $\{\vec{T_i}\}_{i=1}^{N}$
- $L$ has the property that it is large when the network misclassifies.
- An example loss function is MSE:

$$L = \frac{1}{N} \sum_{i=1}^{N} \left( \vec{O_i}(\vec{X_i}, \vec{W}) - \vec{T_i} \right)^2$$

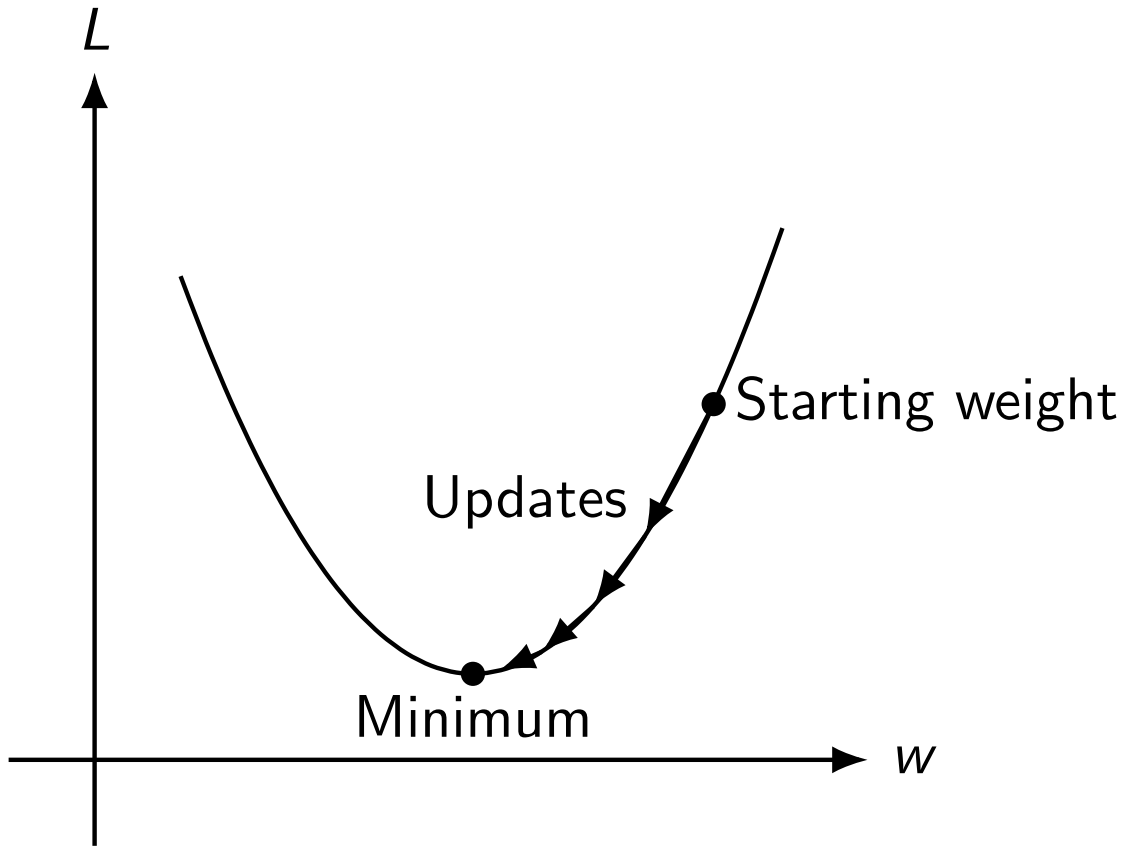
- Our goal is to update the weight vector $\vec{W}$ so that the loss is minimum. That is we want to construct a flow in the weight space which ends at the minima ("**mode**") of the loss function.

# Flow in the weight space



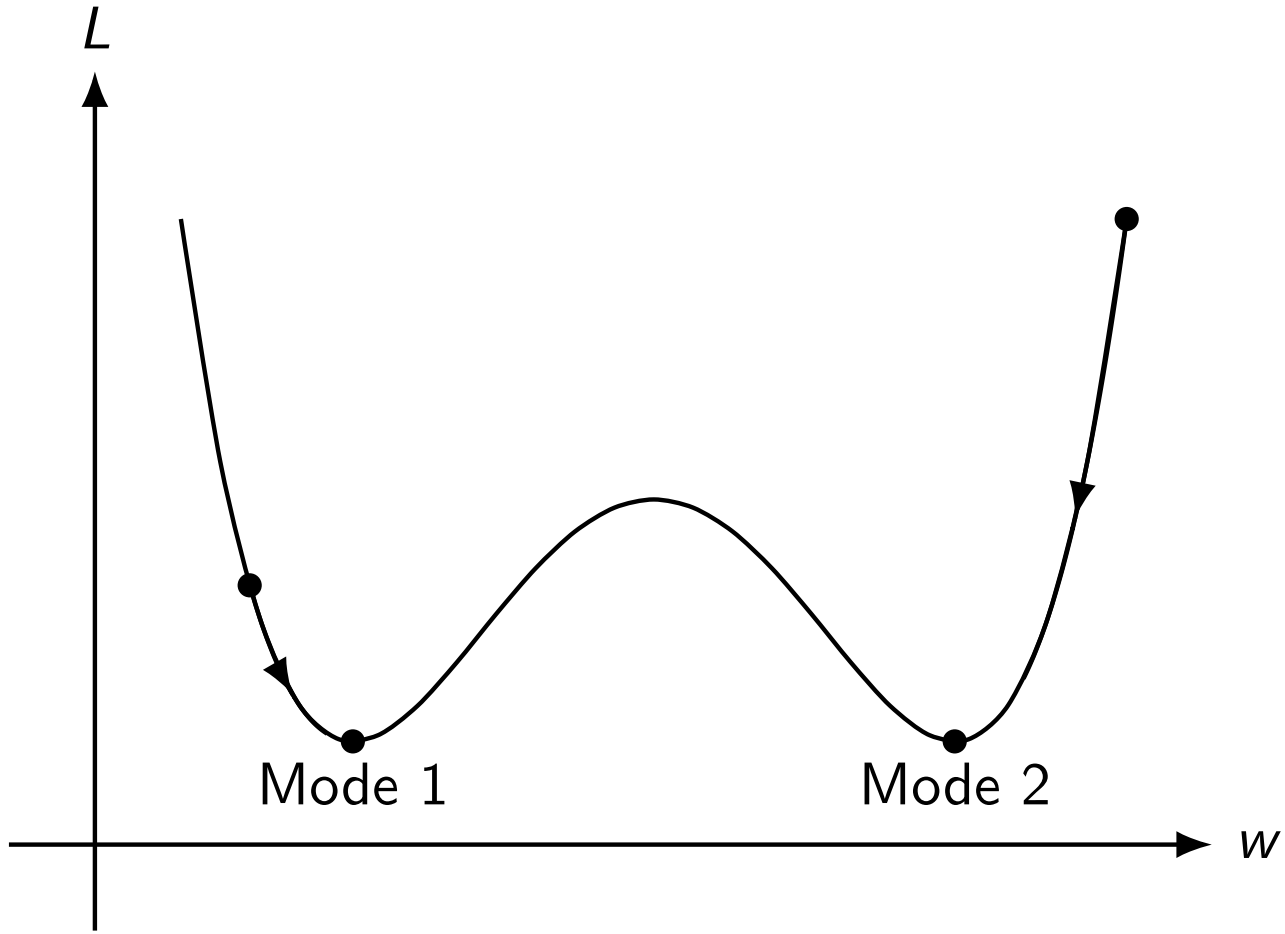
# Example: Loss Function (1D)

- Suppose the model has only one parameter $w$.



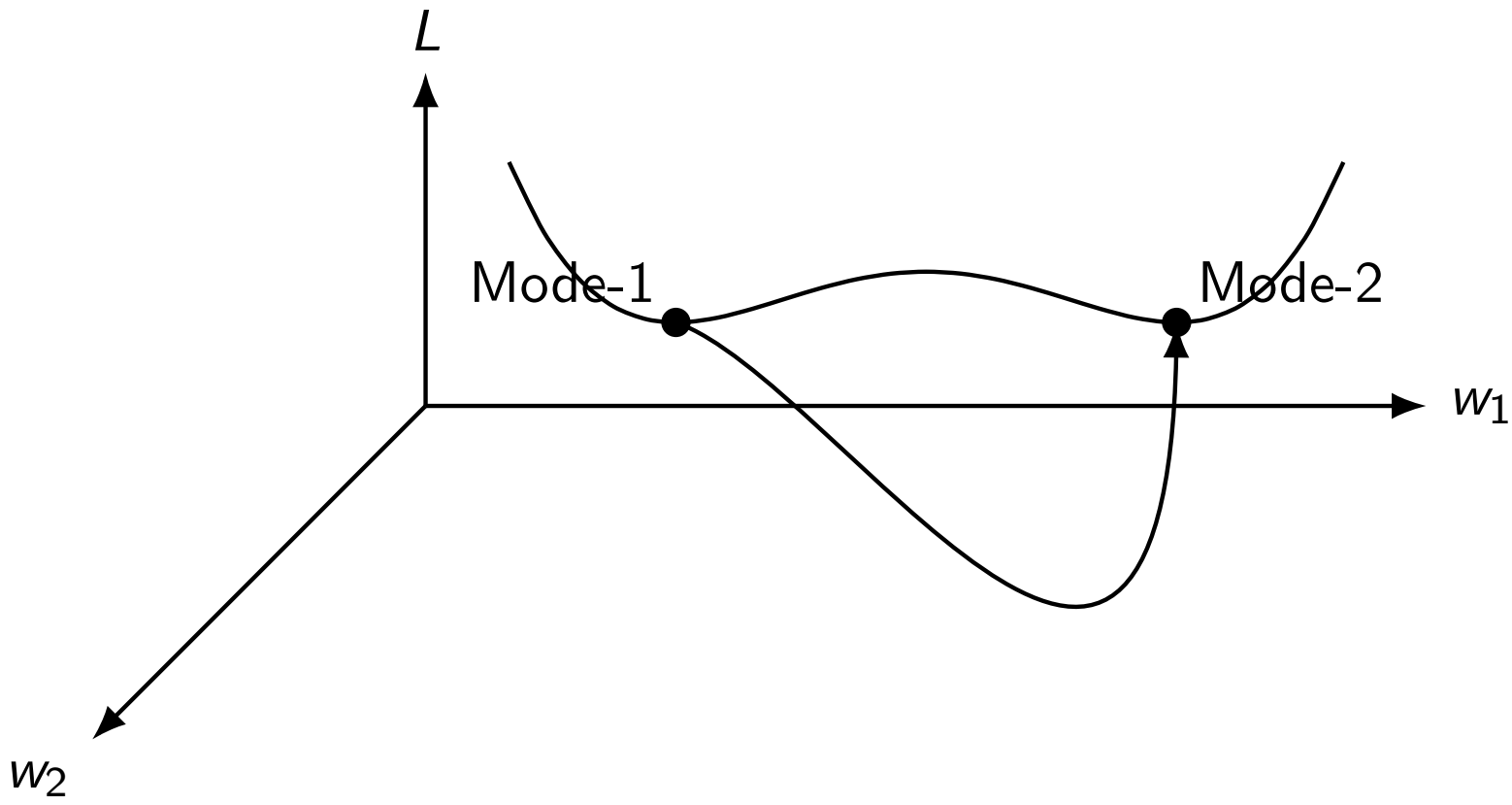
# Loss Function with Multiple Minima

- Example of a loss function with more than one minimum.



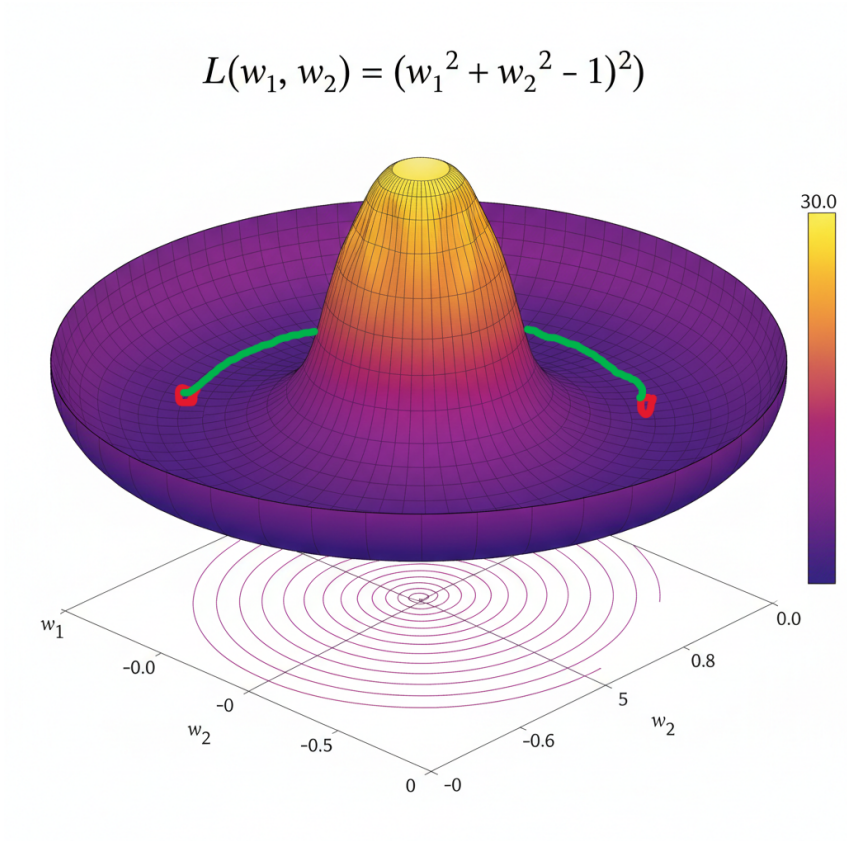

- If we want to go from one minima to the other minima we must go through a loss barrier (that is a path which has high loss in between the minima) since there is only one path in 1D.

# Mode Connectivity in Higher Dimension

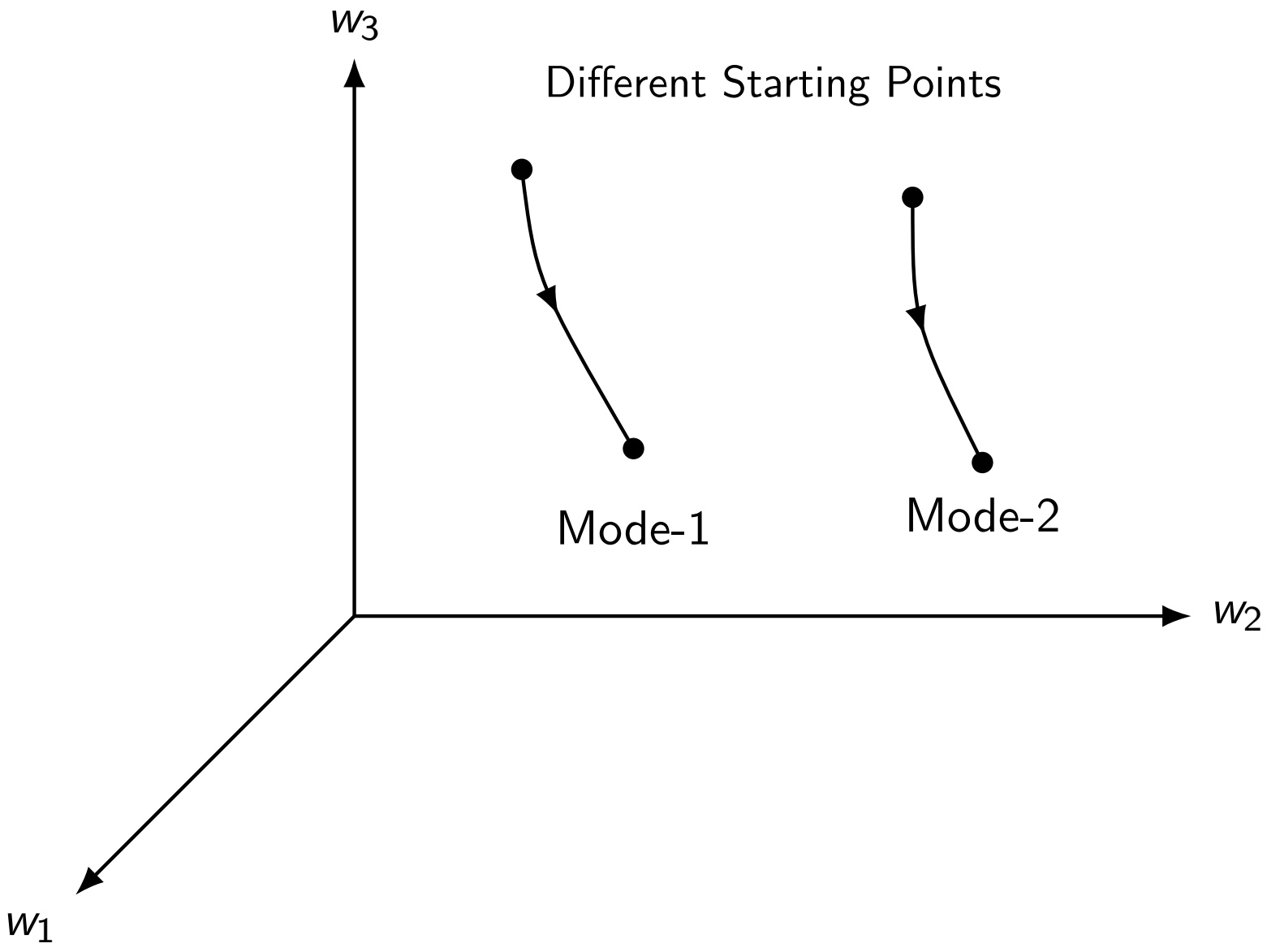



- In higher dimensions there can be paths connecting Mode-1 and Mode-2 along which the loss stays low. This concept is call "**Mode connectivity**"

# Mode connectivity in 2D

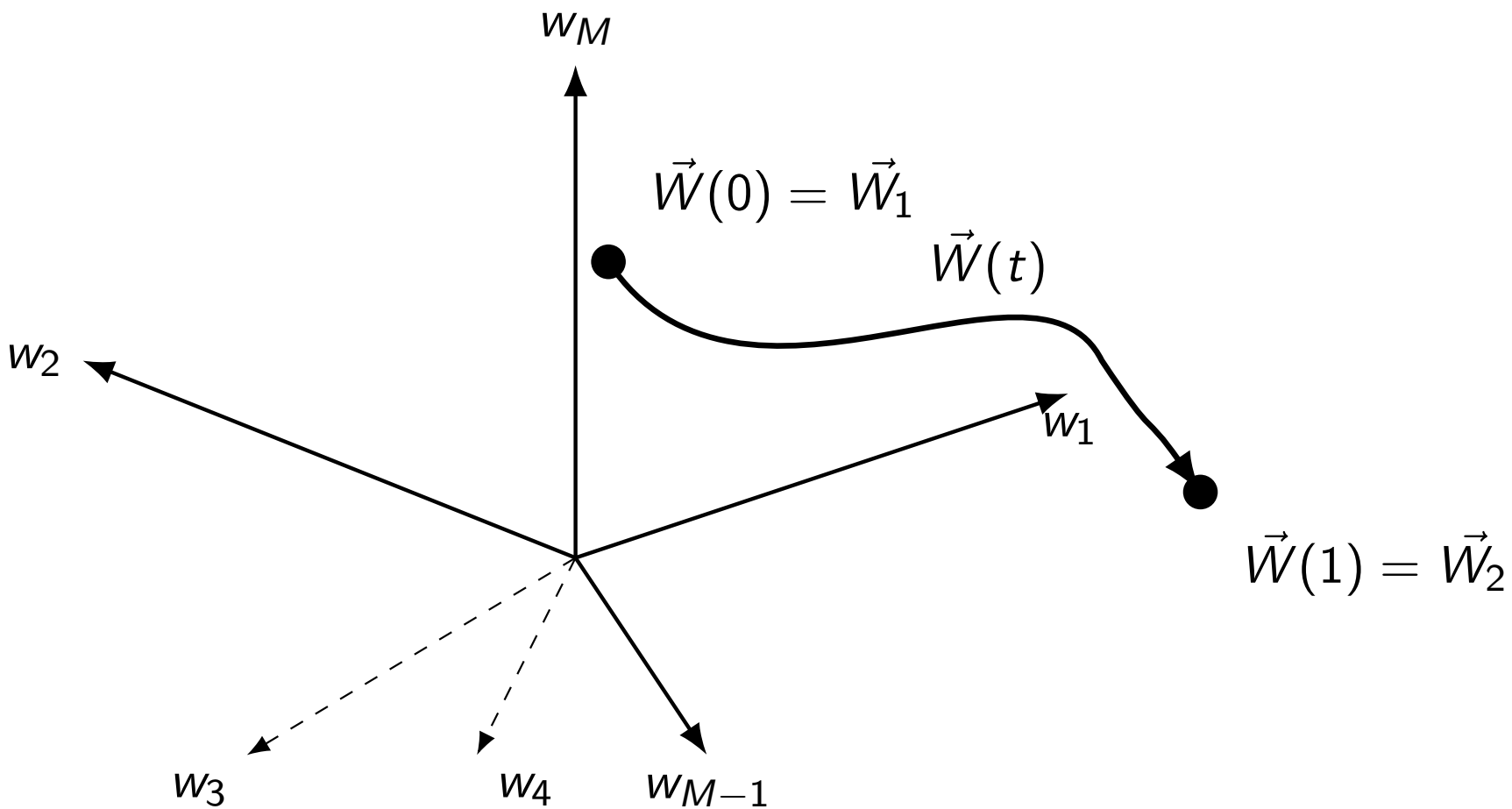


# Flow in the weight space with multiple minima



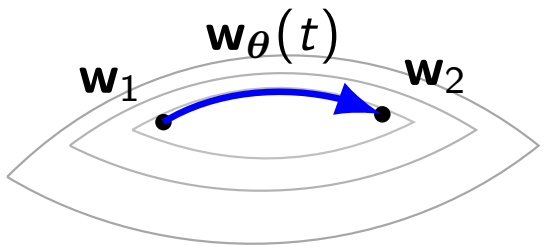
# Constructing a Low-Loss Connecting Path

- Parameters $(w_1, w_2, \ldots, w_M)$
- Suppose we reach at two minima: $\vec{W}_1$ and $\vec{W}_2$ by training the network independently.
- Goal: construct a low-loss curve $\mathbf{w}(t)$ connecting them



# Garipov et al. — Problem Setup

- Let $L(\vec{W})$ be the loss function.
- Let $\vec{W}_{\theta}(t)$ be a curve connecting $\vec{W}_1$ and $\vec{W}_2$.
- $\vec{\theta}$ is a parameter or a set of parameters that controls the shape of the curve.




## Goal

Update the value of $\vec{\theta}$ such that we get a path along the loss remains small.

# Garipov et al. — Optimizing the Path

- Require that the average loss along the curve be minimum.

## Average Loss Along the Curve

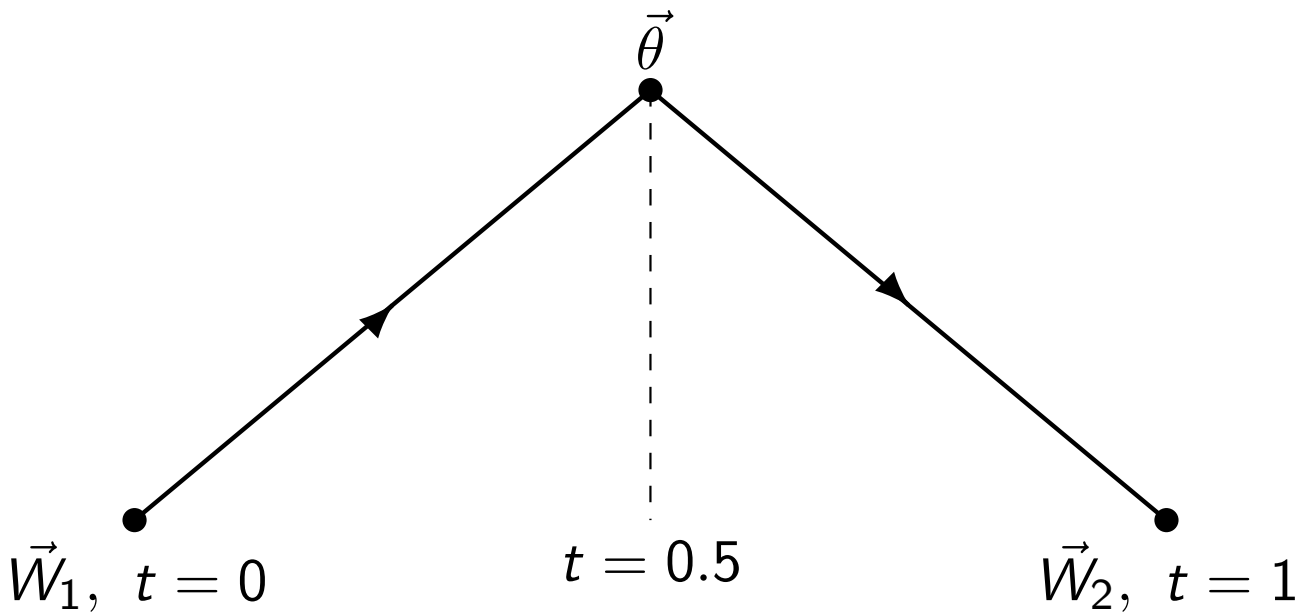
$$\ell(\boldsymbol{\theta}) = \int_0^1 L(\mathbf{w}_{\boldsymbol{\theta}}(t))\, dt$$

## Gradient Update

$$\vec{\boldsymbol{\theta}}_{t+1} = \vec{\boldsymbol{\theta}}_t - \eta\, \vec{\nabla}_{\boldsymbol{\theta}} \ell(\vec{\boldsymbol{\theta}}_t)$$

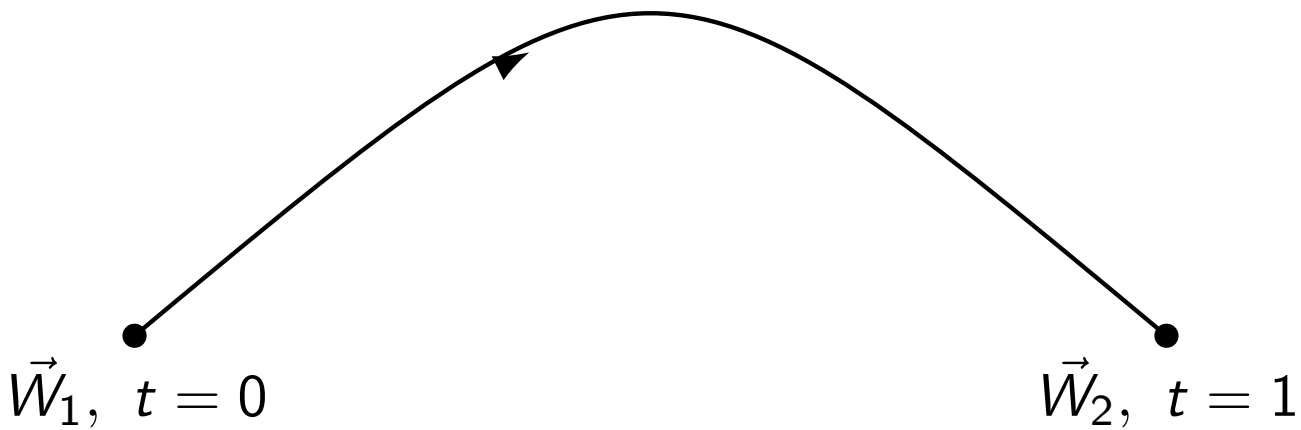
# Example of Paths

$$\vec{W}_\theta(t) = \begin{cases} 2\big(t\vec{\theta} + (0.5 - t)\vec{W}_1\big), & 0 \le t \le 0.5, \\ 2\big((t - 0.5)\vec{W}_2 + (1 - t)\vec{\theta}\big), & 0.5 \le t \le 1. \end{cases}$$

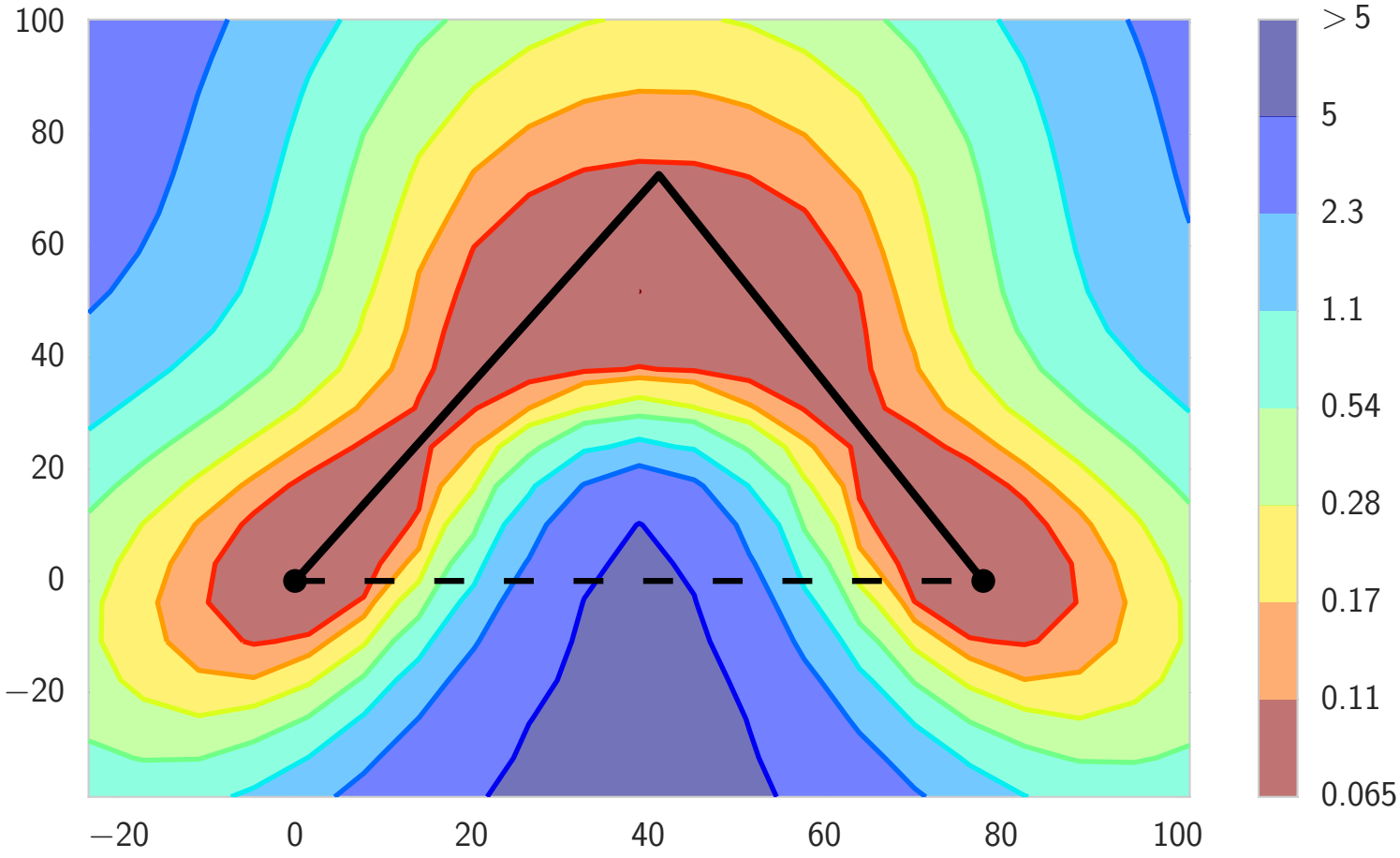


# Bezier Curve

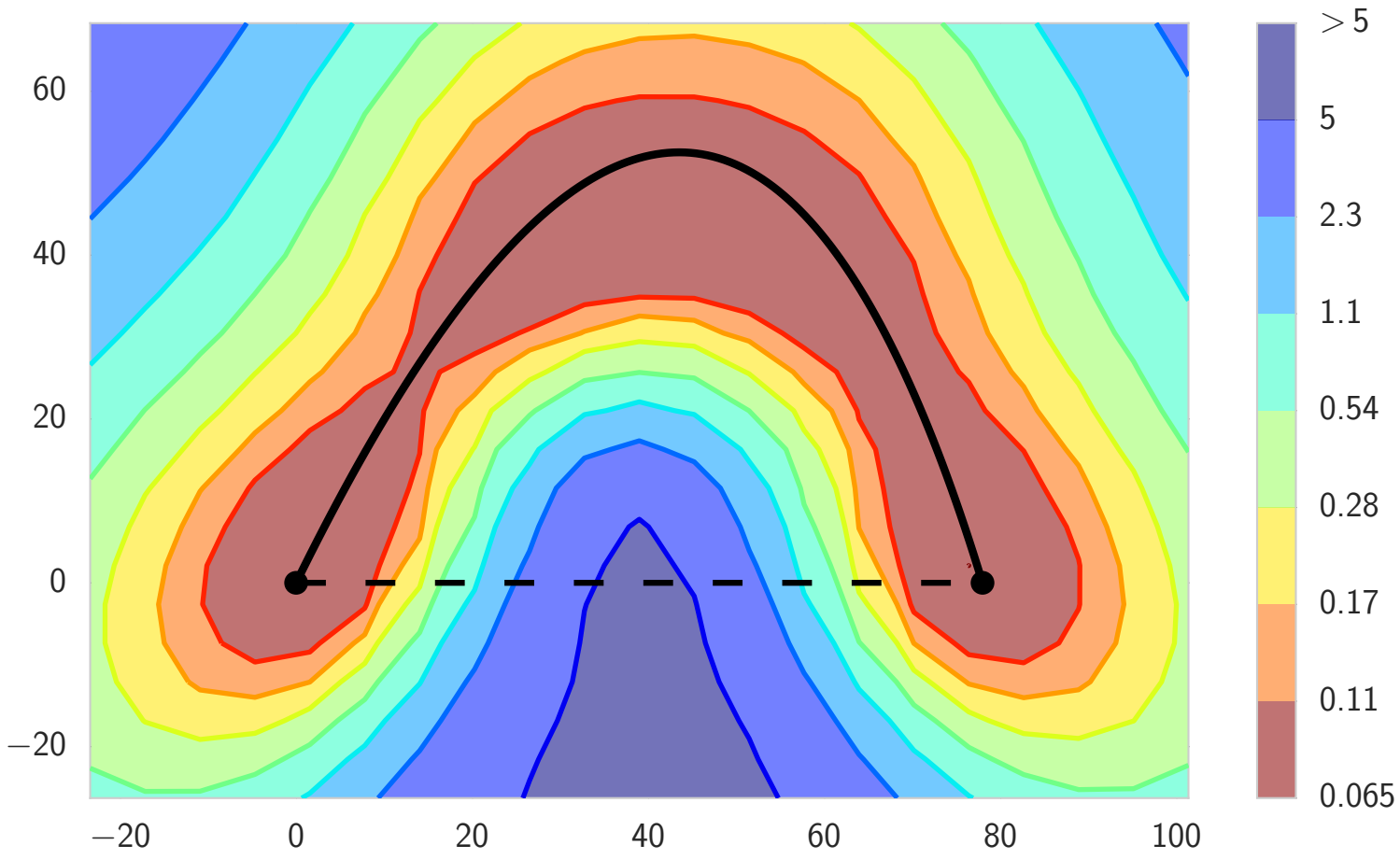
$$\vec{W}_\theta(t) = (1-t)^2\vec{W}_1 + 2t(1-t)\vec{\theta} + t^2\vec{W}_2$$



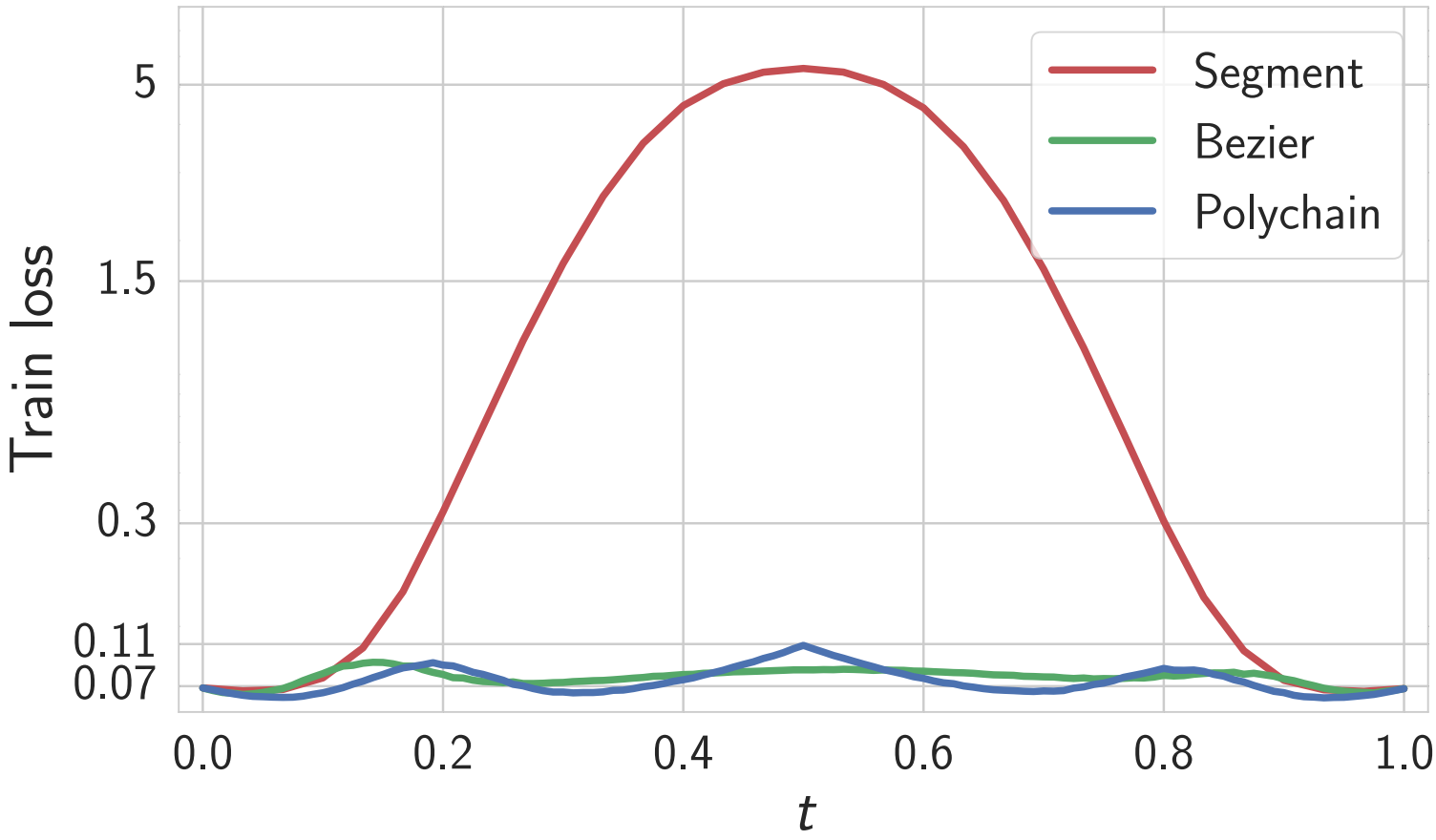
# Results (ResNet-164, CIFAR-100)



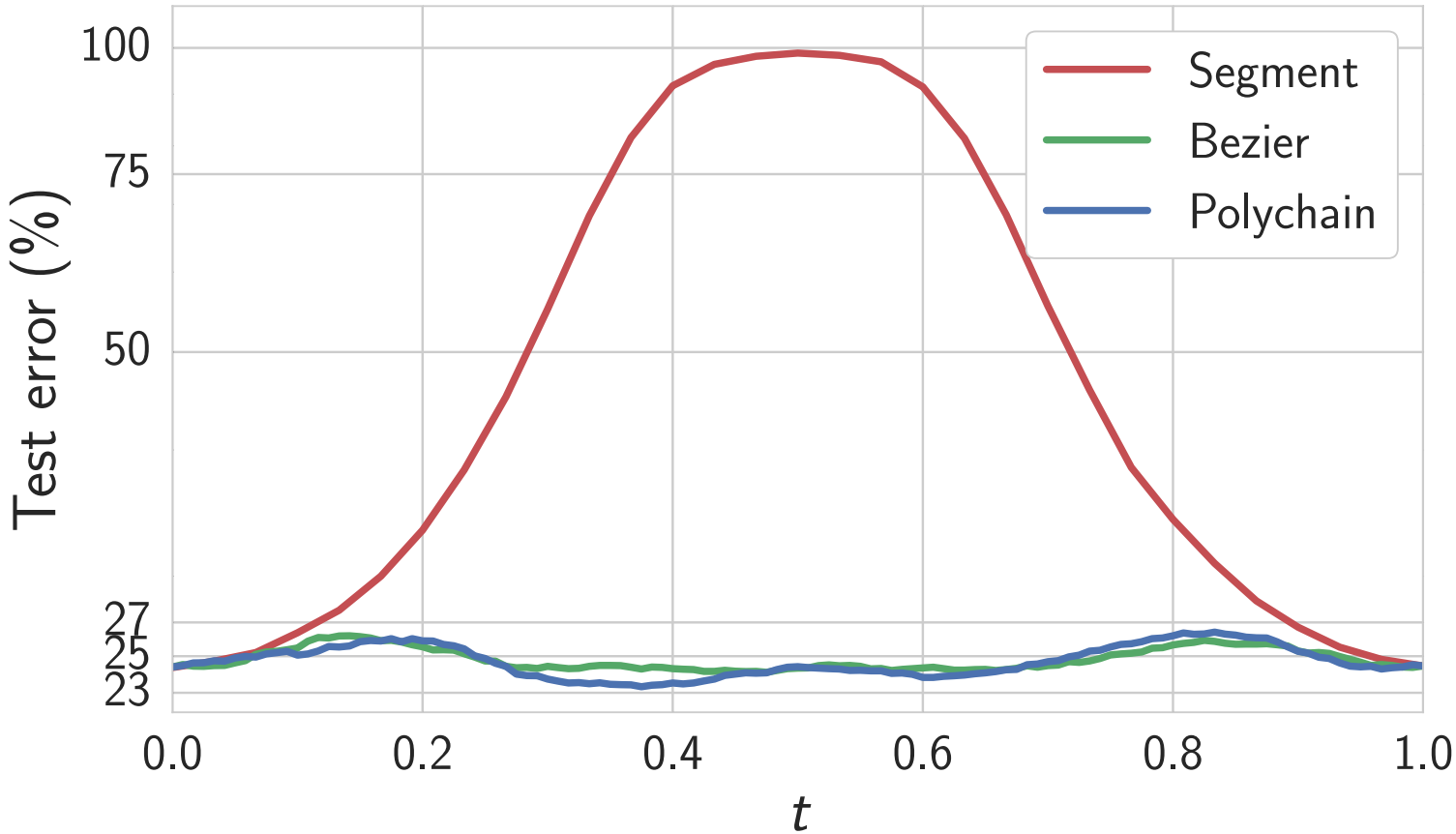
# Results (ResNet-164, CIFAR-100)



# Results(ResNet-164, CIFAR-100)



# Results(ResNet-164, CIFAR-100)

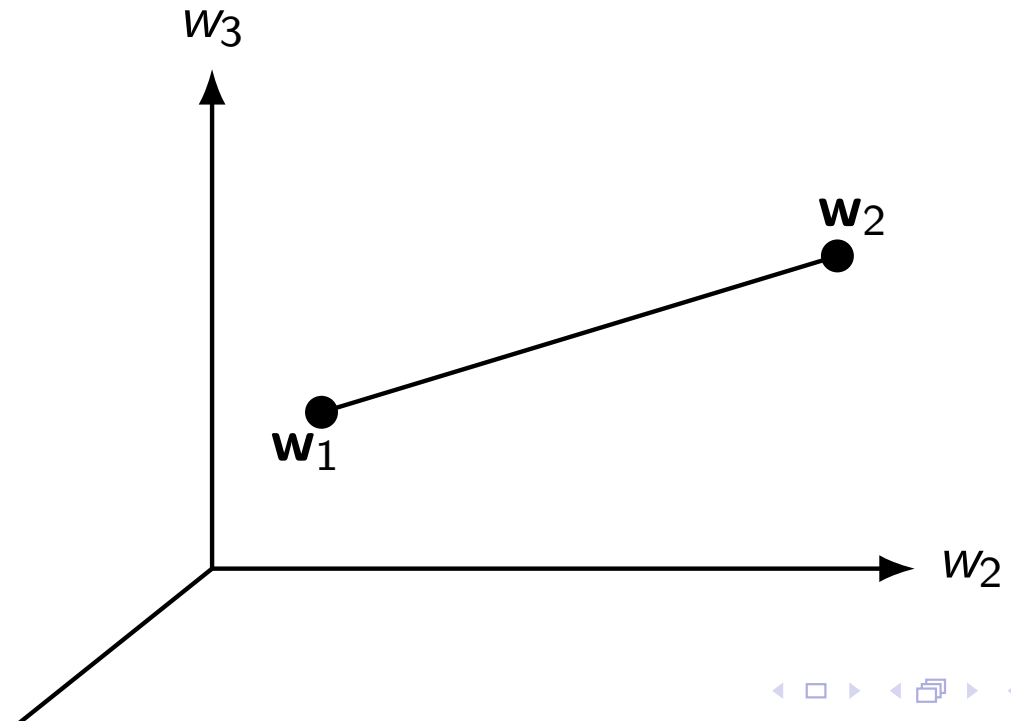


# Draxler et al. — Low-Loss Connectivity

**Goal:** Find a low-loss path between two minima.

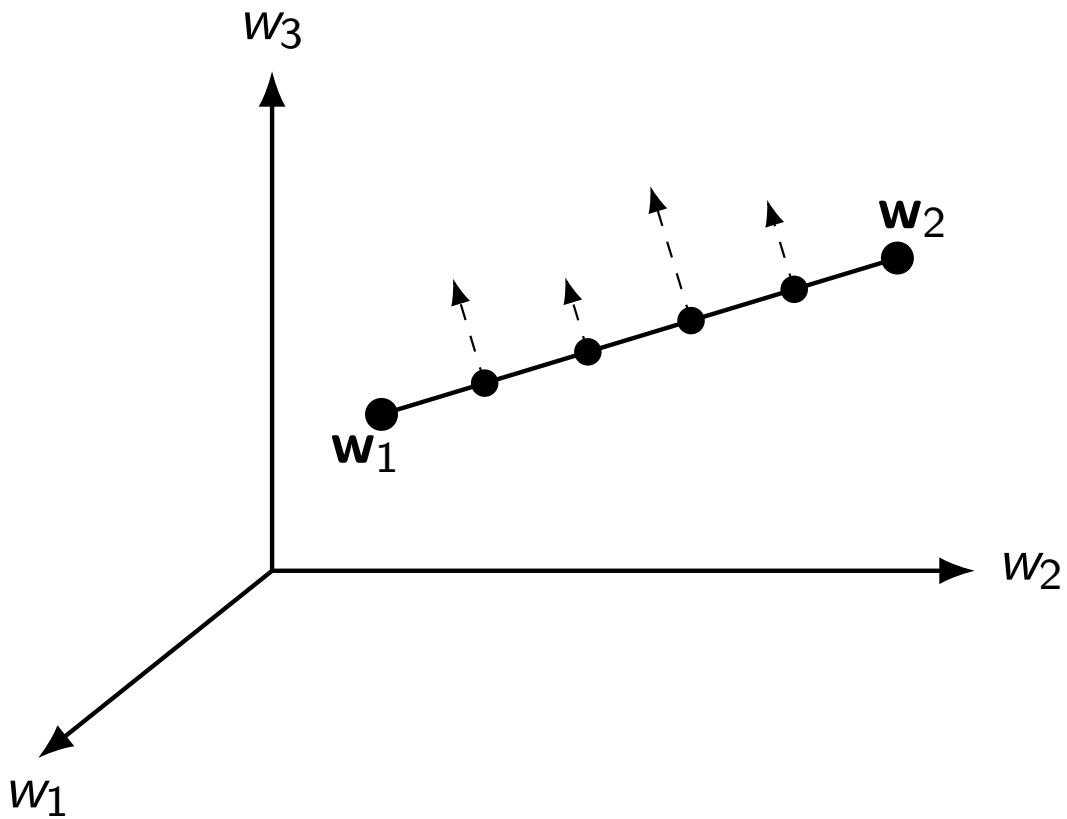
Suppose $\mathbf{w}_1$ and $\mathbf{w}_2$ are two minima. We draw a straight line path between the minimas. We wish to gradually deform the straight line so that we obtain a low loss path.

$$\mathbf{w}(t) = (1-t)\mathbf{w}_1 + t\mathbf{w}_2, \quad t \in [0,1].$$

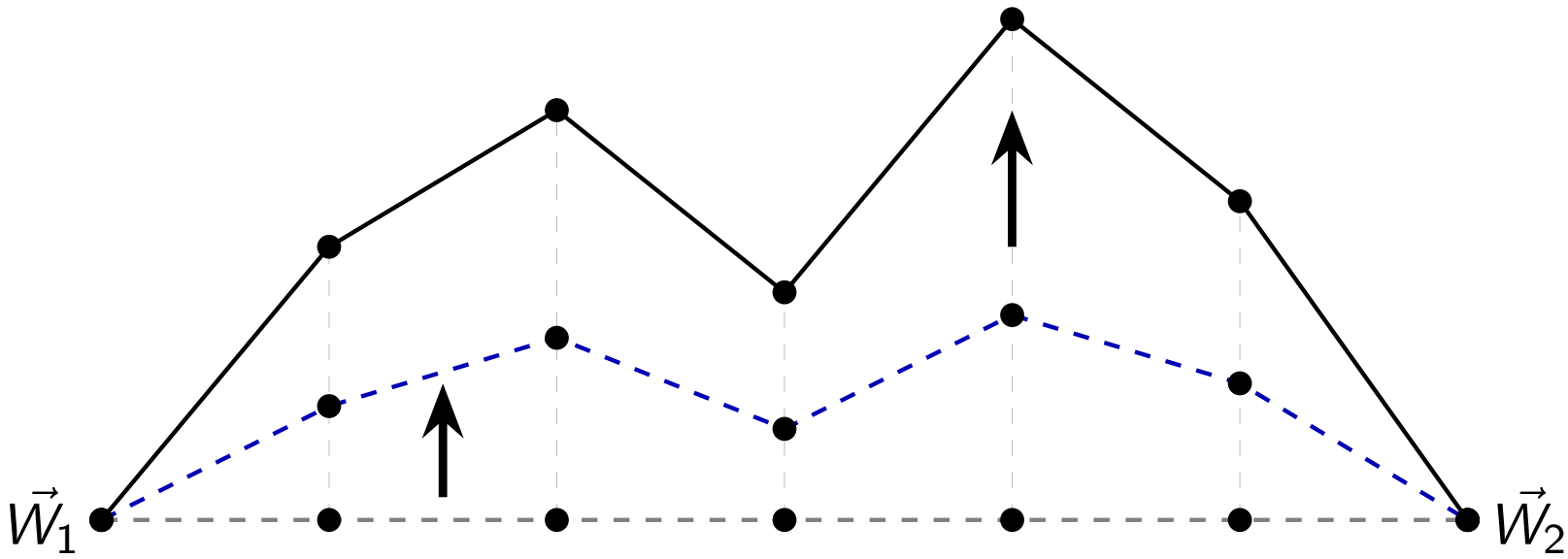


# NEB: Discretization and Perpendicular Updates

We choose points on the line at equal intervals and move the points using some update rule. We make sure that the movements always happen perpendicular to the path since moving tangent to the curve (that is along the curve) does not change the path.



# Deformation of the path



# Update Rule as a Vector Field

**The update rule can be viewed as a vector field F.**

$$\frac{d\vec{w}}{dt} = \vec{F}$$

$$\frac{\vec{W}(t+\Delta t) - \vec{W}(t)}{\Delta t} = \vec{F}$$

$$\Rightarrow \quad \vec{W}(t+\Delta t) = \vec{W}(t) + \Delta t\, \vec{F}$$

Discretizing in time:

$$\vec{W}_{t+1} = \vec{W}_t + \eta \vec{F}$$

$\eta =$ **Learning Rate**

# Vector Field from a Scalar function

**The vector field can be generated by a scalar function $V$:**

$$\mathbf{F} = -\nabla_{\mathbf{w}} V$$

**Choose the potential:**

$$V(\mathbf{w}) = L(\mathbf{w}) + \frac{k}{2} \sum_{i=0}^{N} \|\mathbf{w}_{i+1} - \mathbf{w}_i\|^2$$

**Then the force becomes:**

$$\mathbf{F} = -\nabla_{\mathbf{w}} L(\mathbf{w}) - \frac{k}{2} \sum_{i=0}^{N} \nabla_{\mathbf{w}} \left( \|\mathbf{w}_{i+1} - \mathbf{w}_i\|^2 \right)$$

**Thus F has two terms:**

$$\mathbf{F_L} = -\nabla_{\mathbf{w}} L(\mathbf{w}) \quad \text{(loss term)}$$

$$\mathbf{F}_E = -\frac{k}{2} \sum_{i=0}^{N} \nabla_{\mathbf{w}} \left( \|\mathbf{w}_{i+1} - \mathbf{w}_i\|^2 \right) \quad \text{(elastic term)}$$

# NEB Force Decomposition

**1. Move perpendicular to the curve**

We take only the perpendicular component of the loss term:

$$\mathbf{F_L}_i^{\perp} = -\nabla_{\mathbf{w}_i} L(\mathbf{w}_i) + \left(\nabla_{\mathbf{w}_i} L(\mathbf{w}_i) \cdot \hat{\tau}_i\right)\hat{\tau}_i$$

where $\hat{\tau}_i$ is the unit tangent vector.

We can check that

$$\mathbf{F}_i^{\perp} \cdot \hat{\tau}_i = 0.$$

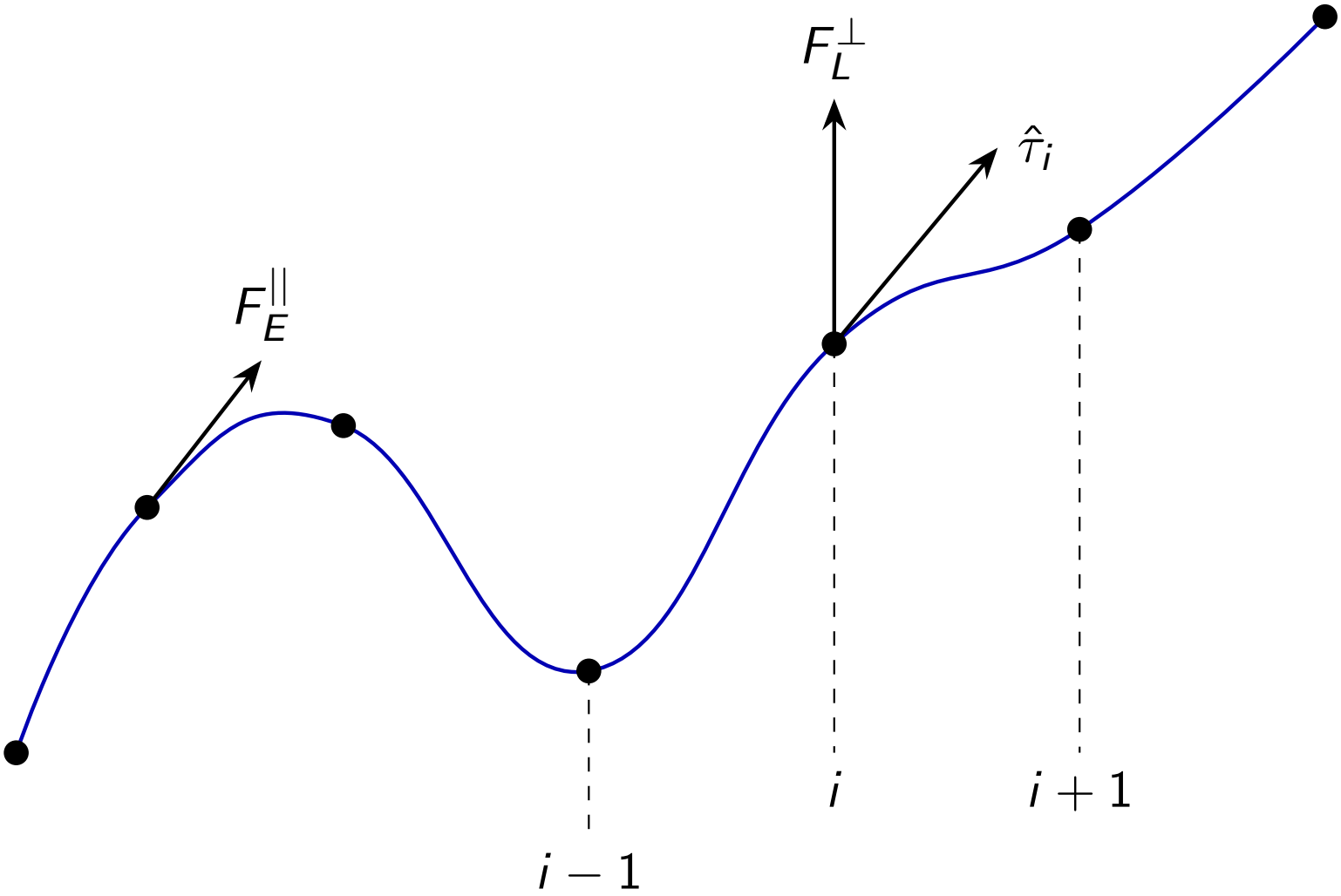
**2. Equal spacing via elastic force**

We keep the *parallel* component of the elastic term:

$$\mathbf{F_E}_i^{\parallel} = k\left(\|\mathbf{w}_{i+1} - \mathbf{w}_i\| - \|\mathbf{w}_i - \mathbf{w}_{i-1}\|\right)\hat{\tau}_i.$$

**Total NEB force:**

$$\mathbf{F}_i = \mathbf{F}_i^{\perp} + \mathbf{F}_i^{\parallel}.$$


$F_L^{\perp}$
$\hat{\tau}_i$
$F_E^{||}$
$i$
$i+1$
$i-1$

# Job of the Elastic Term (1D Illustration)

**Elastic energy:**

$$V_E = \frac{k}{2} \sum_i (w_{i+1} - w_i)^2$$

Focus on the $k$-th point:

$$V_E = \frac{k}{2} \Big[ (w_k - w_{k-1})^2 + (w_{k+1} - w_k)^2 \Big] + \cdots$$

**Derivative with respect to $w_k$:**

$$-\frac{\partial V_E}{\partial w_k} = k(w_{k-1} + w_{k+1} - 2w_k)$$

$$= -k \Big[ (w_k - w_{k-1}) - (w_{k+1} - w_k) \Big]$$

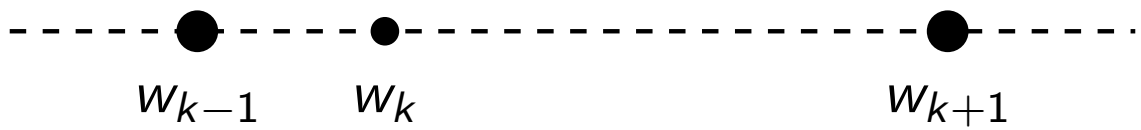
**Update rule:**

$$w_k(t+1) = w_k(t) - \eta k \Big[ (w_k - w_{k-1}) - (w_{k+1} - w_k) \Big]$$

**Interpretation:**
If

$$w_k - w_{k-1} < w_{k+1} - w_k,$$

then the $k$-th point moves to the right (and vice versa).



# Summary of the Process

- We start from a **straight line** with some choice of points at equal intervals.
- We move the points according to the **update rule**:

$$\vec{W}_{i,t+1} = \vec{W}_{i,t} + \eta \left( \vec{F}_{Li}^{\perp} + \vec{F}_{Ei}^{\parallel} \right)$$

- The process stops when:

$$\vec{F}_{Li}^{\perp} = 0 \text{ and } \vec{F}_{Ei}^{\parallel} = 0$$

# Results (DenseNet-40-12, CIFAR-10)