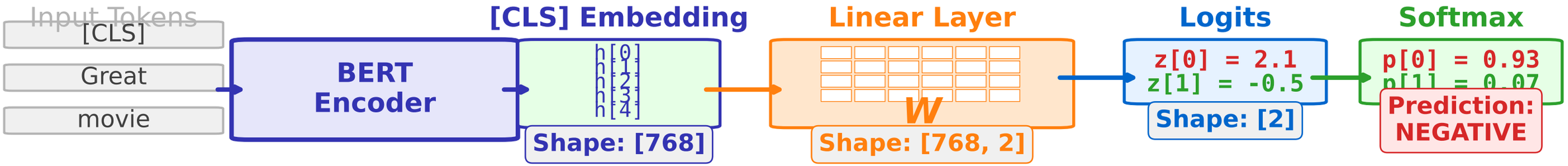
Stage 2: Adding Classifier Head to Pre-trained BERT



Matrix Multiplication Details

```
Linear Layer Computation:

z = W^T · h + b

Where:
  h = [CLS] embedding (768 dimensions)
  W = Weight matrix (768 × 2)
  b = Bias vector (2 dimensions)
  z = Output logits (2 dimensions)

Example (simplified to 3D):
  h = [0.5, -0.2, 0.8]^T
  W = [[0.3, -0.1],
       [0.2,  0.4],
       [-0.1, 0.5]]
  b = [0.1, -0.05]

  z = [0.3*0.5 + 0.2*(-0.2) + (-0.1)*0.8,
       -0.1*0.5 + 0.4*(-0.2) + 0.5*0.8] + b
    = [0.03, 0.27] + [0.1, -0.05]
    = [0.13, 0.22]
```

Initialization Strategy

```
Classifier Head Initialization:

BERT Layers (Stage 1):
  - Load pre-trained weights
  - Already optimized on Wikipedia
  - Frozen or fine-tuned slowly

Linear Layer (Stage 2):
  - Random initialization
  - Xavier/Glorot uniform:
      W ~ U(-√(6/(768+2)), √(6/(768+2)))
  - Bias initialized to zeros: b = [0, 0]

Why Random Init?:
  - No prior knowledge of task
  - Fine-tuning will adapt to sentiment
  - Fast convergence (3-5 epochs)
```