VeriRegime: Distilling High-Performance CNNs to zkML-Optimized MLPs for Trading Signal Generation

DDA4220 Deep Learning - Midterm Report

Lin Boyi 123090327

November 8, 2025

1 Introduction

1.1 Topic and Motivation

Zero-knowledge machine learning (zkML) enables cryptographic verification of neural network inference on blockchain systems, providing trustless and transparent AI decision-making. However, current zkML implementations face significant practical challenges:

- **Computational Overhead**: Complex architectures like CNNs and LSTMs require 100–2000 seconds for proof generation, making real-time applications infeasible.
- **Architecture Constraints**: zkML-friendly operations (e.g., polynomial activations, integer arithmetic) severely limit model expressiveness.
- **Accuracy-Efficiency Trade-off**: Naive simplification of architectures (e.g., direct training of shallow MLPs) often results in 20–40% accuracy degradation.

Research Question: Can we systematically transfer knowledge from high-performance but zkML-unfriendly models (CNNs) to efficient and verifiable models (MLPs) while maintaining competitive accuracy?

Our project proposes a **knowledge distillation framework** that transforms trained CNN models into compact MLPs optimized for zkML deployment. We use cryptocurrency trading signal generation as a testbed, where:

- Time-series financial data provides a realistic benchmark.
- On-chain verifiable trading signals demonstrate practical zkML utility.
- Performance metrics (accuracy, F1 score) are directly interpretable.

1.2 Research Contributions

1. **CNN-MLP Distillation Framework**: Theoretical and empirical analysis of why CNN temporal features can be learned by position-aware MLPs.
2. **zkML Optimization Pipeline**: Combining polynomial activation replacement and Hessian-guided adaptive quantization.
3. **Comprehensive Benchmarking**: End-to-end evaluation of accuracy retention vs. proof generation speedup using the EZKL framework.

2 Related Work

2.1 Knowledge Distillation

Hinton et al. (2015) - Distilling the Knowledge in a Neural Network [1]

The seminal work introduced the concept of "dark knowledge" — soft probability distributions from teacher models that contain richer information than hard labels. Key insights:

- Temperature-scaled softmax ($\sigma_T(z) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$) reveals inter-class similarities.
- Combined loss: $\mathcal{L} = \alpha\mathcal{L}_{\text{CE}}(y, \hat{y}) + \beta\mathcal{L}_{\text{KL}}(p_T, q_T)$.

- Achieves 85–92% teacher accuracy on MNIST/CIFAR-10.

Urban et al. (2017) - Do Deep Convolutional Nets Really Need to be Deep? [2]

Demonstrated that shallow MLPs can mimic deep CNN behavior through distillation:

- Experimented with VGG-16 $\rightarrow$ MLP (CIFAR-10, ImageNet).
- Achieved 87% accuracy retention with 10$\rightarrow$ parameter reduction.
- Revealed that CNNs learn compositional features, but final representations are often linearly separable.

Implications for Our Work:

- CNNs extract local temporal patterns (via convolution), which can be approximated by fully-connected layers given sufficient capacity.
- For 1D time-series, position-dependent weights in MLPs can replicate convolutional feature maps.

2.2 Zero-Knowledge Machine Learning

EZKL (Kang et al., 2023) - ZKML: An Optimizing System for ML Inference in Zero-Knowledge [3]

EZKL compiles ONNX models to arithmetic circuits compatible with Halo2 proof systems:

- Supports operations: polynomial activations, matrix multiplication, quantization.
- Constraint complexity: $C \approx 2^{15}$ to 2^{20} constraints for typical models.
- Proving time: $T \propto C \cdot \log C$ (empirically 10–500 seconds).

Optimizations in Prior Work:

- **Activation Replacement**: Replacing ReLU with x^2, $x - x^3/6$ reduces constraints by 30–50%.
- **Quantization**: Lower bit-width (4–8 bits) reduces field arithmetic operations by 40–60%.

- **Model Simplification**: Pruning, layer fusion, and architecture search.

Gaps Our Work Addresses:

- Existing work focuses on *system-level* optimization (compiler tricks, hardware acceleration).
- We propose *model-level* optimization via distillation to inherently simpler architectures.

2.3 Financial Time-Series Prediction

Reviewed baseline approaches for trading signal generation:

- Technical indicators (EMA, RSI, MACD) as hand-crafted features.
- CNN/LSTM for temporal pattern extraction (accuracy: 55–65% on cryptocurrency data).
- Threshold-based labeling strategies (e.g., ±0.2% price change for BUY/HOLD/SELL).

3 Methodology

3.1 Problem Formulation

Input: Time-series window $\mathbf{X} \in \mathbb{R}^{T \times d}$ ($T = 60$ minutes, $d = 7$ features).
Features: EMA(5,10,20), RSI(14), MACD, VolumeMA(5,10).
Output: Trading signal $y \in \{0, 1, 2\}$ (SELL, HOLD, BUY).
Labeling: Based on 1-hour forward return:

$$r = \frac{p_{t+60} - p_t}{p_t} \times 100\%$$

$$y = \begin{cases} 2 & \text{if } r > 0.2\% \quad \text{(BUY)} \\ 1 & \text{if } -0.2\% \le r \le 0.2\% \quad \text{(HOLD)} \\ 0 & \text{if } r < -0.2\% \quad \text{(SELL)} \end{cases}$$

3.2 CNN Teacher Architecture

Parameters: 27,779 (lightweight for zkML baseline comparison).

Algorithm 1 CNN Teacher Forward Pass

Input: $\mathbf{X} \in \mathbb{R}^{B \times T \times d}$ (batch, time, features)
$\mathbf{X} \leftarrow \text{Permute}(\mathbf{X})$ ▷ $(B, T, d) \rightarrow (B, d, T)$
$\mathbf{H}_1 \leftarrow \text{ReLU}(\text{BN}(\text{Conv1D}_{64}^{k=5}(\mathbf{X})))$
$\mathbf{H}_1 \leftarrow \text{MaxPool}_2(\mathbf{H}_1)$ ▷ $T = 60 \rightarrow 30$
$\mathbf{H}_2 \leftarrow \text{ReLU}(\text{BN}(\text{Conv1D}_{128}^{k=3}(\mathbf{H}_1)))$
$\mathbf{H}_2 \leftarrow \text{MaxPool}_2(\mathbf{H}_2)$ ▷ $T = 30 \rightarrow 15$
$\mathbf{Z} \leftarrow \text{GlobalAvgPool}(\mathbf{H}_2)$ ▷ $(B, 128, 15) \rightarrow (B, 128)$
$\mathbf{Y} \leftarrow \text{Softmax}(\text{FC}_3(\mathbf{Z}))$
Return Y

3.3 MLP Student Architecture (Planned)

Input Flattening: $\mathbf{X}_{\text{flat}} = \text{Flatten}(\mathbf{X}) \in \mathbb{R}^{480}$

Architecture:

$$\mathbf{X}_{\text{flat}} \xrightarrow{\text{FC}_{128}} \mathbf{H}_1 \xrightarrow{\sigma_{\text{poly}}} \mathbf{H}_1' \xrightarrow{\text{FC}_{64}} \mathbf{H}_2 \xrightarrow{\sigma_{\text{poly}}} \mathbf{H}_2' \xrightarrow{\text{FC}_{32}} \mathbf{H}_3 \xrightarrow{\sigma_{\text{poly}}} \mathbf{H}_3' \xrightarrow{\text{FC}_{3}} \mathbf{Y}$$

Parameters: ~80K (similar capacity to CNN but zkML-friendly).

3.4 Knowledge Distillation Loss

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{CE}}(y, \hat{y}_{\text{student}}) + \beta \cdot \mathcal{L}_{\text{KD}}(z_{\text{teacher}}, z_{\text{student}}) + \gamma \cdot \mathcal{L}_{\text{reg}}$$

Where:

- $\mathcal{L}_{\text{CE}}$: Cross-entropy with ground truth labels.
- $\mathcal{L}_{\text{KD}} = \text{KL}(\sigma_T(z_{\text{teacher}}) \| \sigma_T(z_{\text{student}}))$: Distillation loss with temperature T.
- $\mathcal{L}_{\text{reg}} = \lambda_1 \|\mathbf{W}\|_1 + \lambda_2 \sum_i \text{ReLU}(|w_i| - \tau)$: Sparsity regularization.

Hyperparameters to Tune: $\alpha, \beta, T, \lambda_1, \lambda_2, \tau$.

4 Experimental Plan

4.1 Metrics

4.2 Planned Experiments

Success Criteria:

Metric	Symbol	Definition
Accuracy	A	$\frac{\text{Correct Predictions}}{\text{Total Samples}}$
F1 Score	F_1	$\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
Retention Rate	ρ	$\frac{A_{\text{student}}}{A_{\text{teacher}}} \times 100\%$
Constraint Count	C	Number of arithmetic constraints in zkML circuit
Proving Time	T_{prove}	Time to generate zero-knowledge proof (seconds)
Verification Time	T_{verify}	Time to verify proof (milliseconds)
Speedup	S	$\frac{T_{\text{prove}}^{\text{CNN}}}{T_{\text{prove}}^{\text{MLP}}}$

Table 1: Evaluation Metrics

- **Minimum**: $\rho \geq 80\%$, $S \geq 5\times$
- **Expected**: $\rho \geq 85\%$, $S \geq 10\times$, Accuracy drop $< 5\%$
- **Ideal**: $\rho \geq 90\%$, $S \geq 20\times$, Accuracy drop $< 3\%$

5 Progress and Results

5.1 Dataset Construction

Data Source: Binance API (BTC/USDT 1-minute candlesticks)

Time Range: 2022-12-31 to 2024-11-07 (685 days)

Total Samples: 974,907 (after removing NaN from indicator calculations)

Data Split:

- **Training**: 682,434 samples (70%) — 2022-12-31 to 2024-04-18
- **Validation**: 146,236 samples (15%) — 2024-04-18 to 2024-07-29
- **Test**: 146,237 samples (15%) — 2024-07-29 to 2024-11-07

Label Distribution (after threshold optimization):

#	Experiment	Goal	Week
0	CNN Teacher Baseline	Establish performance upper bound	Week 10
1	CNN→MLP Distillation	Validate knowledge transfer	Week 11
2	Polynomial Activation	Reduce constraints while maintaining accuracy	Week 12
3	Adaptive Quantization	Further constraint reduction via Hessian-guided bit-width allocation	Week 12
4	EZKL Compilation	Measure end-to-end zkML performance	Week 13

Table 2: Experiment Roadmap

Label	Training	Validation	Test
SELL (0)	146,917 (21.5%)	36,654 (25.1%)	39,328 (26.9%)
HOLD (1)	374,820 (54.9%)	70,959 (48.5%)	66,233 (45.3%)
BUY (2)	160,637 (23.5%)	38,563 (26.4%)	40,616 (27.8%)

Table 3: Label Distribution Across Splits

Key Challenge Addressed: Initial labeling with ±2% threshold resulted in 99% HOLD labels (severe class imbalance). We systematically analyzed return distributions and optimized the threshold to ±0.2%, achieving a more balanced distribution (23% / 53% / 24%).

5.2 CNN Teacher Training (Experiment 0 - In Progress)

Training Configuration:

- Optimizer: AdamW (lr=1e-3, weight_decay=1e-4)
- Batch Size: 256
- Epochs: 50 (with early stopping, patience=10)
- Loss: CrossEntropy + Label Smoothing (0.1)
- Learning Rate Scheduler: ReduceLROnPlateau (factor=0.5, patience=5)

Current Status (as of Epoch 2/50):

- **Epoch 1 Results**:
  - Train Loss: 0.9627, Train Acc: 56.40%
  - Val Loss: 1.0318, Val Acc: 50.54%
  - **Val F1: 0.3875**
- **Epoch 2** (ongoing): Train Acc trending around 57%

Preliminary Analysis:

- Validation accuracy (50.54%) is only marginally better than random (33.33% for 3-class problem).
- Low F1 score (0.3875) suggests the model struggles with minority classes (SELL/BUY).
- Possible causes:
  1. **Class Imbalance**: Despite threshold optimization, HOLD still dominates (53%).
  2. **Feature Quality**: Technical indicators may not be sufficiently predictive for 1-hour forward returns.
  3. **Model Capacity**: Current CNN (27K params) may be too small.
  4. **Task Difficulty**: Cryptocurrency price movements are inherently noisy and difficult to predict.

5.3 Implementation Details

Code Repository Structure:

```
VeriRegime/
 src/
    data_collection.py      # Binance API data fetching
    data_split.py            # Train/val/test splitting
    relabel_data.py          # Label threshold optimization
    train_cnn.py             # CNN training script
    models/
```

```
      cnn_teacher.py       # CNN architecture
   data/
       dataset.py          # PyTorch Dataset (sliding window)
data/
   btc_usdt_1m_processed.csv
   train.csv, val.csv, test.csv
models/                    # Saved model checkpoints
results/                   # Training logs and visualizations
```

Technical Stack:

- PyTorch 2.9.0
- CCXT (Binance API)
- pandas-ta (technical indicators)
- scikit-learn (metrics)

6 Challenges and Next Steps

6.1 Current Challenges

1. **Low CNN Baseline Performance**:
   - Current validation accuracy (50.54%) is below our target (60%+).
   - This limits the upper bound for distillation experiments.
2. **Class Imbalance**:
   - HOLD class dominates, leading to biased predictions.
   - F1 score (0.3875) indicates poor performance on minority classes.
3. **Feature Engineering**:
   - Current 7 technical indicators may be insufficient.
   - Need to explore additional features (e.g., order book depth, funding rates, volatility metrics).

6.2 Planned Improvements

Short-Term (Week 7-8):

1. **Address Class Imbalance**:
   - Implement class weights in loss function: $\mathcal{L}_{\text{CE}} = -\sum_i w_i y_i \log(\hat{y}_i)$
   - Try focal loss: $\mathcal{L}_{\text{focal}} = -\alpha(1-\hat{y})^\gamma \log(\hat{y})$
   - Oversample minority classes (SMOTE or simple duplication)
2. **Increase Model Capacity**:
   - Add a third convolutional layer ($128 \to 256$ filters)
   - Increase FC layer width ($128 \to 256$)
   - Target: 50–100K parameters (still reasonable for zkML baseline)
3. **Improve Training Strategy**:
   - Longer training (100 epochs instead of 50)
   - Cosine annealing LR schedule
   - Gradient clipping to stabilize training

Medium-Term (Week 8-9):

1. **Feature Engineering**:
   - Add Bollinger Bands, ATR (Average True Range)
   - Include price momentum features (rate of change)
   - Normalize features more carefully (z-score normalization)
2. **Hyperparameter Tuning**:
   - Grid search over learning rates $\{1\text{e}{-4}, 5\text{e}{-4}, 1\text{e}{-3}\}$
   - Experiment with batch sizes $\{128, 256, 512\}$
   - Tune label smoothing $\{0, 0.05, 0.1, 0.2\}$

6.3 Contingency Plans

If CNN baseline remains below 60% after improvements:

- **Option 1**: Accept lower baseline and focus on demonstrating *relative* distillation success (e.g., 85% retention of 55% = 46.75% absolute accuracy).
- **Option 2**: Simplify the task:
  - Binary classification (BUY/SELL, remove HOLD).
  - Predict direction only (up/down) instead of magnitude.
- **Option 3**: Change dataset to a more predictable time-series (e.g., stock market with less volatility, or synthetic data).

6.4 Timeline Adjustments

Week	Status	Tasks
Week 10	Ongoing	CNN baseline training + debugging
Week 11	Planned	MLP Student + Distillation (Exp 1)
Week 12	Planned	Polynomial Activation (Exp 2) + Quantization (Exp 3)
Week 13	Planned	EZKL Compilation (Exp 4)
Week 14–15	Planned	Analysis, Pareto optimization, report writing
Week 16	Planned	Final report submission

Table 4: Updated Timeline

7 Conclusion

This midterm report documents the progress of the VeriRegime project, which aims to optimize neural networks for zero-knowledge machine learning through knowledge distillation. We have:

- Established a solid theoretical foundation by reviewing distillation and zkML literature.
- Collected and preprocessed a large-scale financial time-series dataset (975K samples).
- Implemented and begun training a CNN Teacher baseline model.

Key Findings So Far:

1. Label threshold optimization is critical for balanced classification in financial data.
2. Initial CNN performance (50.54% validation accuracy) is below target, indicating room for improvement.
3. The project is technically feasible but requires iterative refinement.

Next Steps:

1. Address class imbalance through weighted loss functions.
2. Increase model capacity and training duration.
3. Proceed with distillation experiments once a satisfactory baseline is achieved.

We remain confident that the proposed distillation framework will demonstrate meaningful improvements in zkML efficiency while maintaining competitive accuracy for trading signal generation.

References

[1] Hinton, G., Vinyals, O., & Dean, J. (2015). *Distilling the knowledge in a neural network*. arXiv preprint arXiv:1503.02531.

[2] Urban, G., Geras, K. J., Kahou, S. E., Aslan, O., Wang, S., Caruana, R., ... & Bengio, Y. (2017). *Do deep convolutional nets really need to be deep and convolutional?*. ICLR 2017.

[3] Kang, D., Hashimoto, T., Stoica, I., & Sun, Y. (2023). *ZKML: An optimizing system for ML inference in zero-knowledge*. Cryptology ePrint Archive.

[4] Dong, Z., Yao, Z., Gholami, A., Mahoney, M. W., & Keutzer, K. (2019). *HAWQ: Hessian aware quantization of neural networks with mixed-precision*. ICCV 2019.

[5] Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., & Keutzer, K. (2021). *A survey of quantization methods for efficient neural network inference*. arXiv preprint arXiv:2103.13630.