Storm Damage Classification in Switzerland

End Term Project Neural Networks NAIL002

Nils Gämperli, 20.1.2024

Idea

SWI swissinfo.ch

Swiss perspectives in 10 languages

Climate change ›

How to protect people and places from landslides in Switzerland

- swissinfo.ch (accessed 15.1.2025)

1°

Blick

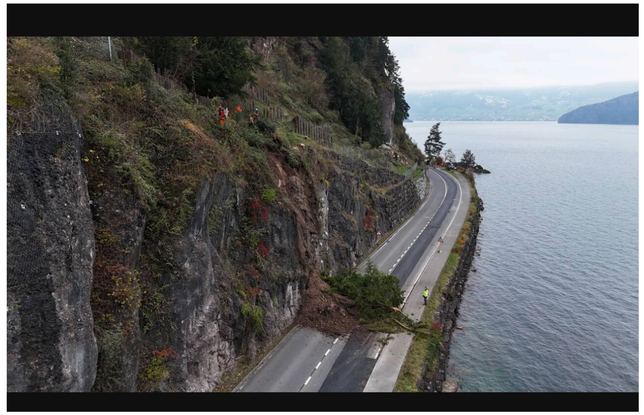
Erdrutsch in Weggis LU

Experten kämpfen gegen instabilen Hang

Nach dem Erdrutsch in Weggis LU droht ein weiterer Felssturz. Die Strasse zwischen Brunnen und Küssnacht SZ bleibt vorerst gesperrt. Spezialisten kämpfen gegen einen instabilen Felsbrocken im Hang.

Publiziert: 30.11.2024 um 10:45 Uhr | Aktualisiert: 30.11.2024 um 11:22 Uhr

4



blick.ch (accessed 15.1.25)

Idea

How likely is a storm damage to happen on a specific date?

Initial Data

- Storm Damage Database
- Total of 32 dimensions (coordinates, municipalities….)
- Classification of the damage
  - Small: [10’000 ; 400’000[CHF
  - Medium: [400’000 - 2’000’000[CHF
  - Large: > 2’000’000 or death
- Main Process: Water/Mudslide, Landslide, Collapse

Idea Adaption

A storm damage has just happened, how large is the damage?

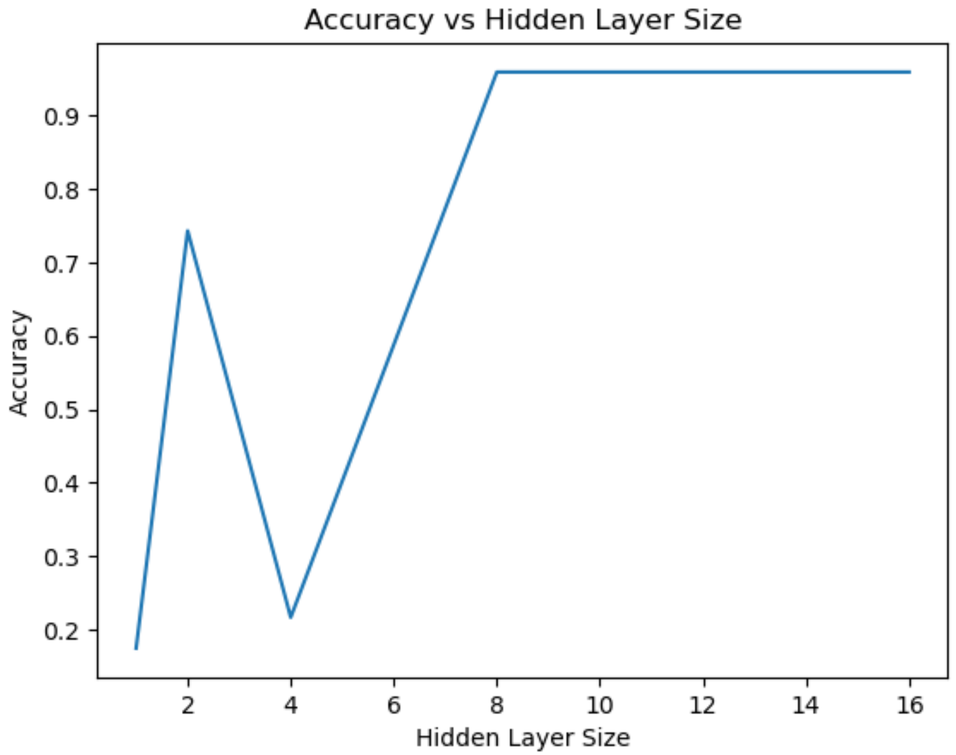
Data Preparation

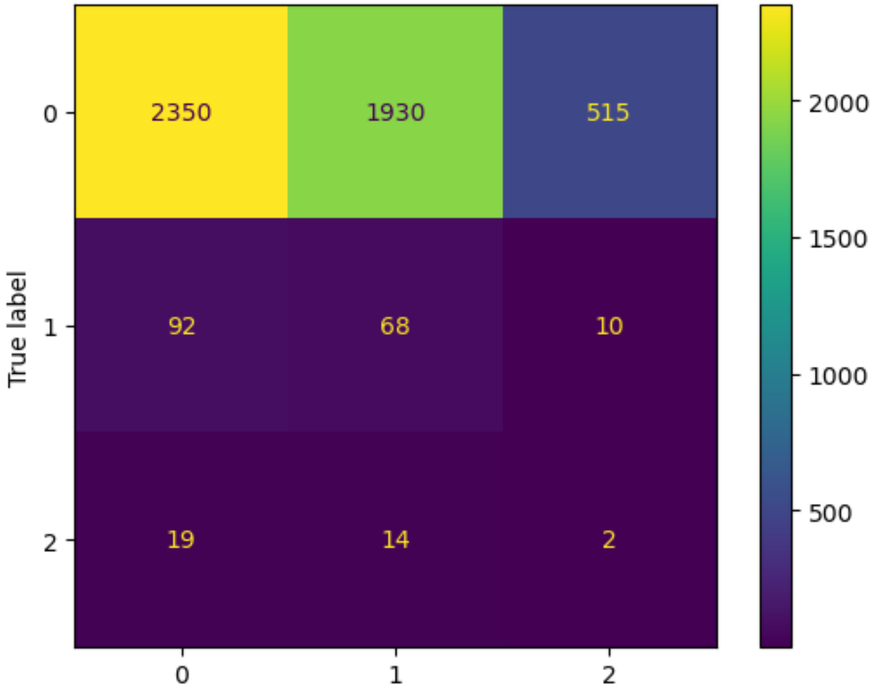
- Kept the following dimension: coordinates, date, Main process, damage classification (desired output)
- Mapped to daily weather data of previous 7 days from OpenMeteo API
  - Max temperature
  - Min temperature
  - Sunshine duration
  - Rain
  - Snowfall

Architecture of the Neural Network

- Framework: PyTorch
- Activation Function: RelU
- Loss Function: Cross Entropy Loss
- Optimizer: Adam (Combination of RMSprop and Momentum)
- Input Layer Size: 37
- Output Layer Size: 3

One Hidden Layer





Difficulties

- Imbalanced Dataset
- Occurrence of Classes (Damage Size)
  - Small: 4740
  - Medium: 214
  - Large: 45

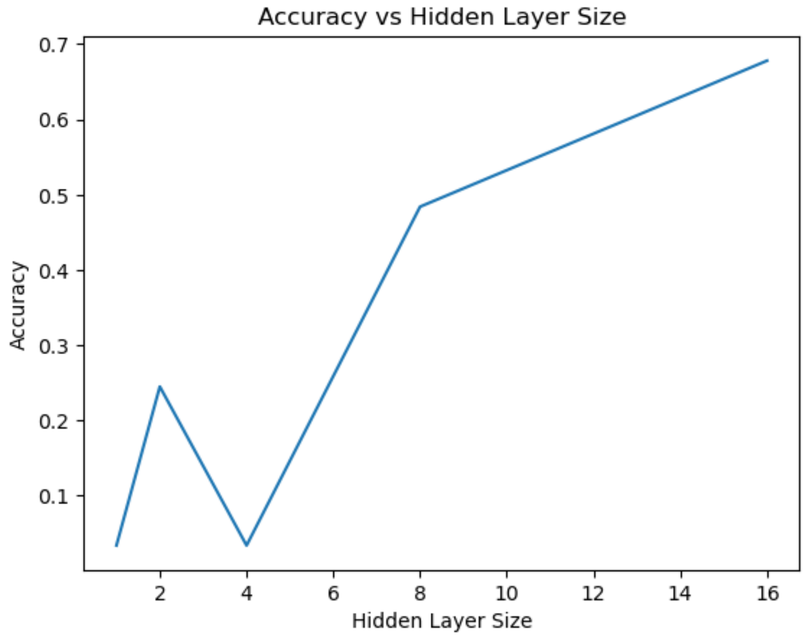
Weighted Classes

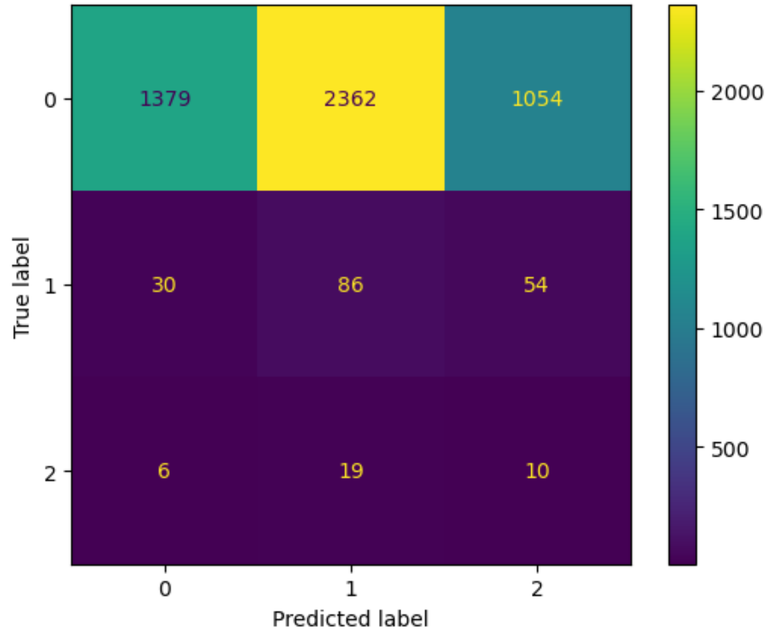
- Punish errors on minority classes harder
- Easy to implement with PyTorch

```
class_counts = torch.bincount(y_train_tensor)
class_weights = 1.0 / class_counts.float()
class_weights = class_weights / class_weights.sum()
class_weights = class_weights.to(torch.float32)
……
criterion = nn.CrossEntropyLoss(weight=class_weights)
```

Weighted Classes

- Punish errors on minority classes harder





Conclusion

- Imbalanced Data set leads to difficulties
- Include more data (not only weather)
- Use a bigger Neural Network
- Training on a subset
- Complex Domain

Questions?