Whose Inflation Rates Matter Most?

A DSGE Model & ML Approach to Monetary Policy in the Euro Area

OeNB & SUERF Annual Economic Conference 2024

Daniel Stempel **Johannes Zahner**

Heinrich-Heine-Universität Düsseldorf

Goethe University Frankfurt

June 2024

Table of Contents

Introduction
Inflation differentials

- Euro area monetary policy is conducted uniformly for 20 member countries.
- How does the ECB react to deflationary *and* inflationary pressure in member states?
- Particularly relevant if countries deviate structurally from the euro area average inflation rate.

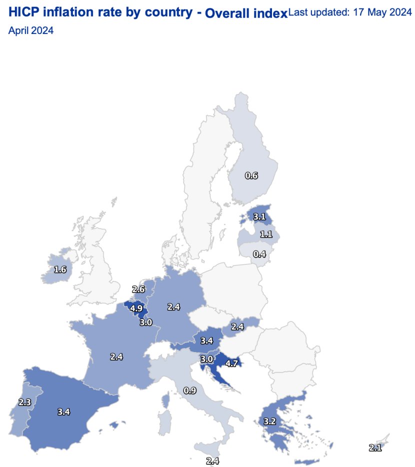


Figure. Source: ECB Website

Introduction
Inflation differentials

- Euro area monetary policy is conducted uniformly for 20 member countries.
- How does the ECB react to deflationary *and* inflationary pressure in member states?
- Particularly relevant if countries deviate structurally from the euro area average inflation rate.

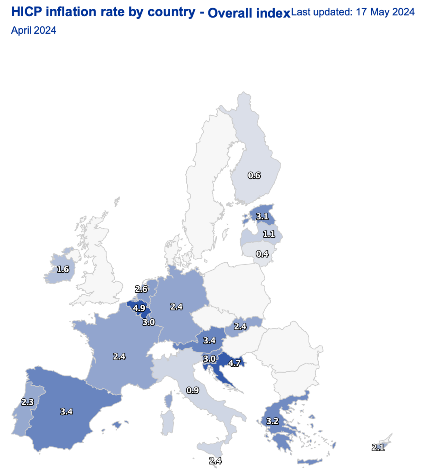


Figure. Source: ECB Website

Introduction
Inflation differentials

- Euro area monetary policy is conducted uniformly for 20 member countries.
- How does the ECB react to deflationary *and* inflationary pressure in member states?
- Particularly relevant if countries deviate structurally from the euro area average inflation rate.

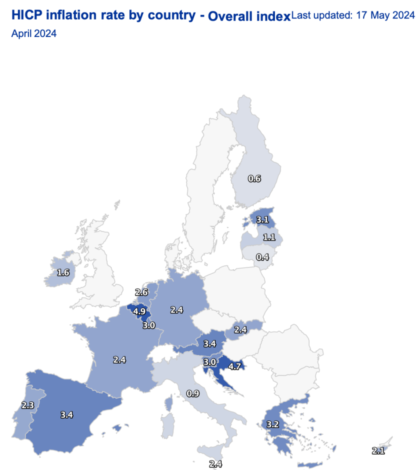


Figure. Source: ECB Website

Introduction

Inflation Development in the Euro Area

- EMU-members structurally differ in the volatility of their inflation rates.
  - Austria, Germany, the Netherlands → Low Volatility
  - Greece, Ireland, Italy, Portugal, and Spain → High Volatility

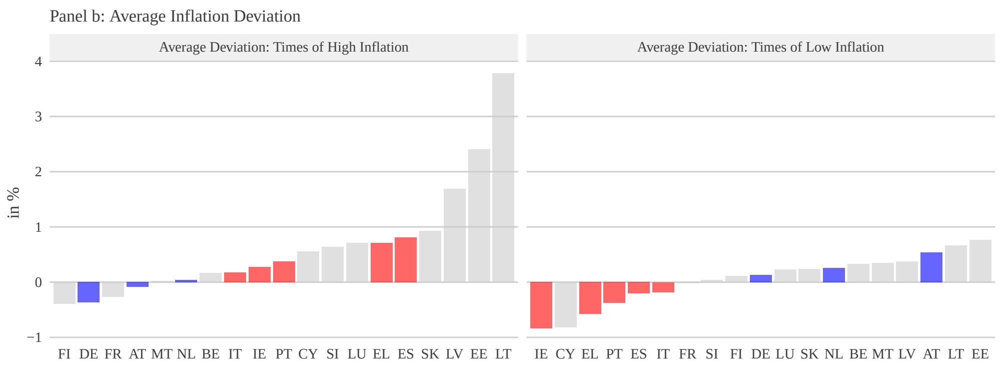


Figure. Average Inflation Deviations.

Introduction

Calculating the bias?

Whose inflation rates matter most for the ECB's monetary policy?

Table. EMU Taylor Rule

	Dependent variable:				
	Interest Rate				
	(1)	(2)	(3)	(4)	(5)
HICP -2%	2.04^{***}	2.24^{***}	2.52^{***}	2.44^{***}	1.86^{***}
	(0.12)	(0.14)	(0.19)	(0.26)	(0.32)
Constant	−0.21	−0.17	−0.12	−0.20	-0.41^{*}
	(0.16)	(0.17)	(0.19)	(0.22)	(0.24)
Weight on LV countries (ω) =	0	0.2	0.5	0.8	1
Observations	240	240	240	240	240
R^2	0.56	0.53	0.43	0.26	0.12

Note: HICP is calculated as follows: $HICP := \omega \times CPI_{LV} + (1-\omega) \times CPI_{HV}$.

→ Which weight accurately describes historical EMU monetary policy?

Introduction

Calculating the bias?

Whose inflation rates matter most for the ECB's monetary policy?

Table. EMU Taylor Rule

	Dependent variable:				
	Interest Rate				
	(1)	(2)	(3)	(4)	(5)
HICP -2%	2.04^{***}	2.24^{***}	2.52^{***}	2.44^{***}	1.86^{***}
	(0.12)	(0.14)	(0.19)	(0.26)	(0.32)
Constant	−0.21	−0.17	−0.12	−0.20	-0.41^{*}
	(0.16)	(0.17)	(0.19)	(0.22)	(0.24)
Weight on LV countries (ω) =	0	0.2	0.5	0.8	1
Observations	240	240	240	240	240
R^2	0.56	0.53	0.43	0.26	0.12

Note: HICP is calculated as follows: $HICP := \omega \times CPI_{LV} + (1-\omega) \times CPI_{HV}$.

→ Which weight accurately describes historical EMU monetary policy?

Introduction

Calculating the bias?

Whose inflation rates matter most for the ECB's monetary policy?

Table. EMU Taylor Rule

	Dependent variable:				
	Interest Rate				
	(1)	(2)	(3)	(4)	(5)
HICP -2%	2.04^{***}	2.24^{***}	2.52^{***}	2.44^{***}	1.86^{***}
	(0.12)	(0.14)	(0.19)	(0.26)	(0.32)
Constant	−0.21	−0.17	−0.12	−0.20	-0.41^{*}
	(0.16)	(0.17)	(0.19)	(0.22)	(0.24)
Weight on LV countries (ω) =	0	0.2	0.5	0.8	1
Observations	240	240	240	240	240
R^2	0.56	0.53	0.43	0.26	0.12

Note: HICP is calculated as follows: $HICP := \omega \times CPI_{LV} + (1 - \omega) \times CPI_{HV}$.

→ Which weight accurately describes historical EMU monetary policy?

Introduction

Paper overview

Our "data-driven" solution:

1. Build a two-country New Keynesian model (NKM) of a monetary union with different central bank regimes.
2. Simulate NKM to generate a data for different policy regime.
3. Train machine learning models (ML) in classifying each regimes.
4. Use the trained ML to classify historical EMU data.

Results:

1. Distribution of the ECB's historical inflation weight is biased.
2. ECB react more strongly to countries whose inflation rates exhibit larger deviations from their long-term trend.

Introduction
Paper overview

Our "data-driven" solution:

1. Build a two-country New Keynesian model (NKM) of a monetary union with different central bank regimes.
2. Simulate NKM to generate a data for different policy regime.
3. Train machine learning models (ML) in classifying each regimes.
4. Use the trained ML to classify historical EMU data.

Results:

1. Distribution of the ECB's historical inflation weight is biased.
2. ECB react more strongly to countries whose inflation rates exhibit larger deviations from their long-term trend.

Introduction

Paper overview

Our "data-driven" solution:

1. Build a two-country New Keynesian model (NKM) of a monetary union with different central bank regimes.
2. Simulate NKM to generate a data for different policy regime.
3. Train machine learning models (ML) in classifying each regimes.
4. Use the trained ML to classify historical EMU data.

Results:

1. Distribution of the ECB's historical inflation weight is biased.
2. ECB react more strongly to countries whose inflation rates exhibit larger deviations from their long-term trend.

Introduction

Paper overview

Our "data-driven" solution:

1. Build a two-country New Keynesian model (NKM) of a monetary union with different central bank regimes.
2. Simulate NKM to generate a data for different policy regime.
3. Train machine learning models (ML) in classifying each regimes.
4. Use the trained ML to classify historical EMU data.

Results:

1. Distribution of the ECB's historical inflation weight is biased.
2. ECB react more strongly to countries whose inflation rates exhibit larger deviations from their long-term trend.

Introduction

Paper overview

Our "data-driven" solution:

1. Build a two-country New Keynesian model (NKM) of a monetary union with different central bank regimes.
2. Simulate NKM to generate a data for different policy regime.
3. Train machine learning models (ML) in classifying each regimes.
4. Use the trained ML to classify historical EMU data.

Results:

1. Distribution of the ECB's historical inflation weight is biased.
2. ECB react more strongly to countries whose inflation rates exhibit larger deviations from their long-term trend.

Introduction

Paper overview

Our "data-driven" solution:

1. Build a two-country New Keynesian model (NKM) of a monetary union with different central bank regimes.
2. Simulate NKM to generate a data for different policy regime.
3. Train machine learning models (ML) in classifying each regimes.
4. Use the trained ML to classify historical EMU data.

Results:

1. Distribution of the ECB's historical inflation weight is biased.
2. ECB react more strongly to countries whose inflation rates exhibit larger deviations from their long-term trend.

Table of Contents

Combining DSGE and Machine Learning

- Simple currency union with 2 countries (*HV* and *LV*) with each a household and a firm sector
- Monetary Policy Targeting Rule:

$$i_t = \rho + \phi_\pi \left(\omega_\pi \pi_t^{C,HV} + (1 - \omega_\pi) \pi_t^{C,LV} \right)$$

  1. central bank reacts to the union-wide inflation rate: $\omega_\pi \approx 0.5$
  2. central bank reacts more strongly to country *HV*: $\omega_\pi = 0.8$
  3. central bank reacts more strongly to country *LV*: $\omega_\pi = 0.2$
- Calibrate country *LV* (*HV*) to represent the *LV* (*HV*) EMU-members[1]
- Simulate 3×10.000 periods of macro variables $(C, L, \pi, ...)$ $\rightarrow$ Split train/test set: 80/20
- Train/Evaluate ML models on the simulated data (neural network outperforms the other models).
- Use neural network to predict ω on historical EMU data.

→ ML evaluation → Model fit

[1] Breuss and Rabitsch (2008) for AT, Albonico et al. (2019) for DE, and Garcia et al. (2021) for NL; Papageorgiou (2014) (EL), Garcia et al. (2021) (IE), Albonico et al. (2019) (ES, IT), and Almeida (2009) (PT)

Combining DSGE and Machine Learning

- Simple currency union with 2 countries (*HV* and *LV*) with each a household and a firm sector
- Monetary Policy Targeting Rule:

$$i_t = \rho + \phi_\pi \left(\omega_\pi \pi_t^{C,HV} + (1 - \omega_\pi) \pi_t^{C,LV} \right)$$

1. central bank reacts to the union-wide inflation rate: $\omega_\pi \approx 0.5$
2. central bank reacts more strongly to country *HV*: $\omega_\pi = 0.8$
3. central bank reacts more strongly to country *LV*: $\omega_\pi = 0.2$

- Calibrate country *LV* (*HV*) to represent the *LV* (*HV*) EMU-members[1]
- Simulate 3×10.000 periods of macro variables ($C, L, \pi, \ldots$) → Split train/test set: 80/20
- Train/Evaluate ML models on the simulated data (neural network outperforms the other models).
- Use neural network to predict ω on historical EMU data.

→ ML evaluation → Model fit

[1] Breuss and Rabitsch (2008) for AT, Albonico et al. (2019) for DE, and Garcia et al. (2021) for NL; Papageorgiou (2014) (EL), Garcia et al. (2021) (IE), Albonico et al. (2019) (ES, IT), and Almeida (2009) (PT)

Combining DSGE and Machine Learning

- Simple currency union with 2 countries (*HV* and *LV*) with each a household and a firm sector
- Monetary Policy Targeting Rule:

$$i_t = \rho + \phi_\pi \left(\omega_\pi \pi_t^{C,HV} + (1 - \omega_\pi)\, \pi_t^{C,LV} \right)$$

  1. central bank reacts to the union-wide inflation rate: $\omega_\pi \approx 0.5$
  2. central bank reacts more strongly to country *HV*: $\omega_\pi = 0.8$
  3. central bank reacts more strongly to country *LV*: $\omega_\pi = 0.2$
- Calibrate country *LV* (*HV*) to represent the *LV* (*HV*) EMU-members[1]
- Simulate 3×10.000 periods of macro variables $(C, L, \pi, ...)$ → Split train/test set: 80/20
- Train/Evaluate ML models on the simulated data (neural network outperforms the other models).
- Use neural network to predict ω on historical EMU data.

→ ML evaluation → Model fit

[1] Breuss and Rabitsch (2008) for AT, Albonico et al. (2019) for DE, and Garcia et al. (2021) for NL; Papageorgiou (2014) (EL), Garcia et al. (2021) (IE), Albonico et al. (2019) (ES, IT), and Almeida (2009) (PT)

Combining DSGE and Machine Learning

- Simple currency union with 2 countries (*HV* and *LV*) with each a household and a firm sector
- Monetary Policy Targeting Rule:

$$i_t = \rho + \phi_\pi \left(\omega_\pi \pi_t^{C,HV} + (1 - \omega_\pi) \pi_t^{C,LV} \right)$$

  1. central bank reacts to the union-wide inflation rate: $\omega_\pi \approx 0.5$
  2. central bank reacts more strongly to country *HV*: $\omega_\pi = 0.8$
  3. central bank reacts more strongly to country *LV*: $\omega_\pi = 0.2$
- Calibrate country *LV* (*HV*) to represent the *LV* (*HV*) EMU-members[1]
- Simulate 3×10.000 periods of macro variables $(C, L, \pi, ...)$ → Split train/test set: 80/20
- Train/Evaluate ML models on the simulated data (neural network outperforms the other models).
- Use neural network to predict ω on historical EMU data.

→ ML evaluation → Model fit

[1] Breuss and Rabitsch (2008) for AT, Albonico et al. (2019) for DE, and Garcia et al. (2021) for NL; Papageorgiou (2014) (EL), Garcia et al. (2021) (IE), Albonico et al. (2019) (ES, IT), and Almeida (2009) (PT)

Combining DSGE and Machine Learning

- Simple currency union with 2 countries (HV and LV) with each a household and a firm sector
- Monetary Policy Targeting Rule:

$$i_t = \rho + \phi_\pi \left(\omega_\pi \pi_t^{C,HV} + (1 - \omega_\pi) \pi_t^{C,LV} \right)$$

1. central bank reacts to the union-wide inflation rate: $\omega_\pi \approx 0.5$
2. central bank reacts more strongly to country HV: $\omega_\pi = 0.8$
3. central bank reacts more strongly to country LV: $\omega_\pi = 0.2$

- Calibrate country LV (HV) to represent the LV (HV) EMU-members[1]
- Simulate 3×10.000 periods of macro variables $(C, L, \pi, ...)$ $\rightarrow$ Split train/test set: 80/20
- Train/Evaluate ML models on the simulated data (neural network outperforms the other models).
- Use neural network to predict ω on historical EMU data.

→ ML evaluation → Model fit

[1] Breuss and Rabitsch (2008) for AT, Albonico et al. (2019) for DE, and Garcia et al. (2021) for NL; Papageorgiou (2014) (EL), Garcia et al. (2021) (IE), Albonico et al. (2019) (ES, IT), and Almeida (2009) (PT)

Combining DSGE and Machine Learning

- Simple currency union with 2 countries (HV and LV) with each a household and a firm sector
- Monetary Policy Targeting Rule:

$$i_t = \rho + \phi_\pi \left(\omega_\pi \pi_t^{C,HV} + (1 - \omega_\pi)\, \pi_t^{C,LV} \right)$$

  1. central bank reacts to the union-wide inflation rate: $\omega_\pi \approx 0.5$
  2. central bank reacts more strongly to country HV: $\omega_\pi = 0.8$
  3. central bank reacts more strongly to country LV: $\omega_\pi = 0.2$
- Calibrate country LV (HV) to represent the LV (HV) EMU-members[1]
- Simulate 3×10.000 periods of macro variables $(C, L, \pi, ...)$ → Split train/test set: 80/20
- Train/Evaluate ML models on the simulated data (neural network outperforms the other models).
- Use neural network to predict ω on historical EMU data.

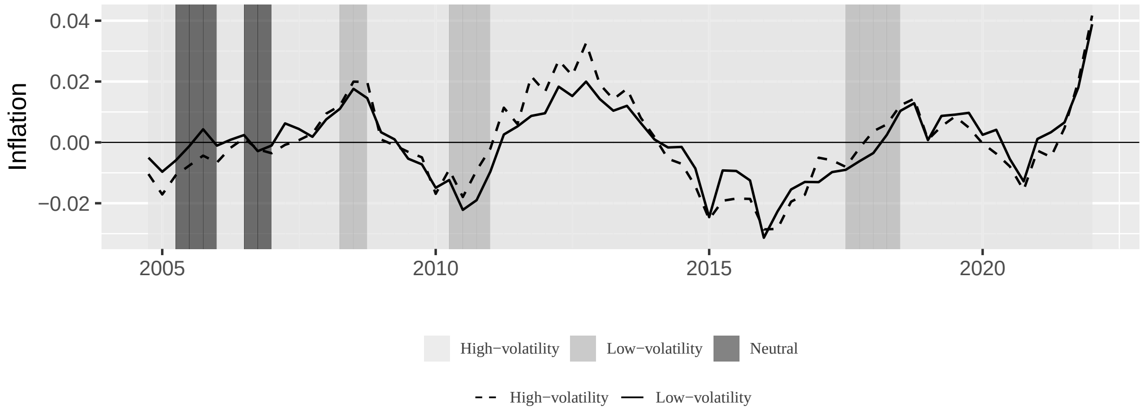
→ ML evaluation → Model fit

[1] Breuss and Rabitsch (2008) for AT, Albonico et al. (2019) for DE, and Garcia et al. (2021) for NL; Papageorgiou (2014) (EL), Garcia et al. (2021) (IE), Albonico et al. (2019) (ES, IT), and Almeida (2009) (PT)

Table of Contents

Results

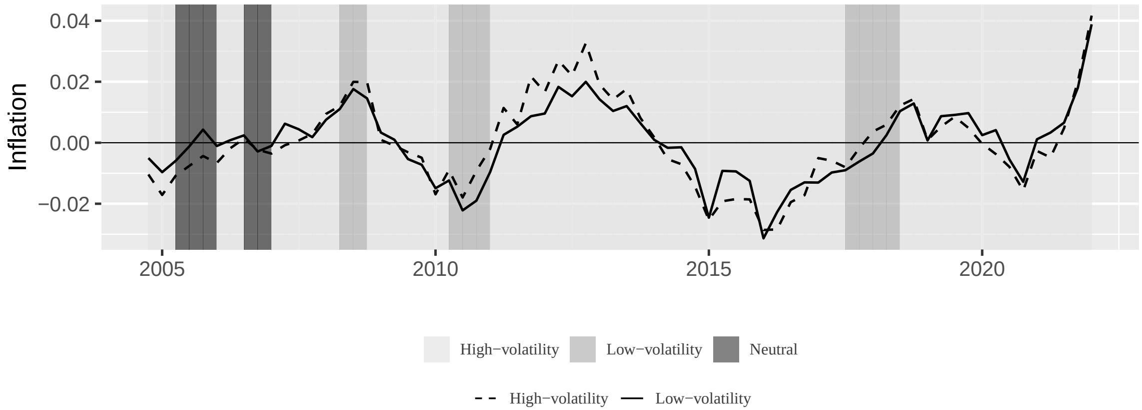
Monetary Policy Regime Classifications



1. Biased weight: disproportional emphasis (80%) on *HV* inflation rates
2. ECB is reacting more strongly to greater deviations of inflation rates from their long-term trend → potential explanation for 1.

Results

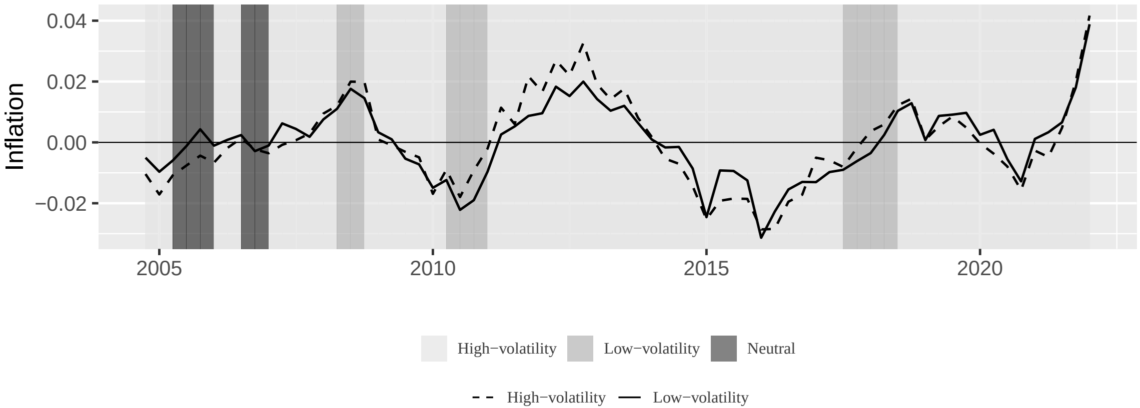
Monetary Policy Regime Classifications



1. Biased weight: disproportional emphasis (80%) on *HV* inflation rates
2. ECB is reacting more strongly to greater deviations of inflation rates from their long-term trend → potential explanation for 1.

Results

Monetary Policy Regime Classifications



1. Biased weight: disproportional emphasis (80%) on *HV* inflation rates
2. ECB is reacting more strongly to greater deviations of inflation rates from their long-term trend → potential explanation for 1.

Results

On the ECB's Taylor Rule and Loss Function

Standard central bank loss function:

$$L_t = -\frac{1}{2}\left(\pi_t^{EMU}\right)^2$$

where π_t^{EMU} is the EMU-wide inflation rate. The corresponding Taylor rule is given by:

$$i_t = \rho + \phi_\pi \pi_t^{EMU}.$$

Results

On the ECB's Taylor Rule and Loss Function

If ECB's losses arise from individual deviations rather than from aggregated ones:

$$L_t = -\frac{1}{2}\sum_{k=1}^{K}\omega^k\left(\pi_t^k\right)^2$$

The interest rate rule becomes:

$$i_t = \rho + \phi_\pi\left(\sum_{k=1}^{K}\Omega_t^k\pi_t^k\right)$$
$$\Omega_t^k = \omega^k - \nu\left(|\pi_t^{EMU}| - |\pi_t^k|\right)$$

Example:

- *HV* inflation deviation is greater than *LV*'s ($|\pi_t^{HV}| > |\pi_t^{EMU}|$)
- *HV* weight in the Taylor Rule exceeds the "true" *HV* weight: $\Omega_t^{HV} > \omega^{HV}$.

Results

Regression Model

- Problem: We require weights in continuous space
- Adjustments:
    1. NKM: redefine the inflation weight: $\Omega_\pi \in [0.1, 0.9]$
    2. Simulate the NKM in 0.1 Ω_π increments
    3. Regression NN
- Repeat Training and evaluation of NN
- (As expected:) biased weight (0.67) favors the high-volatility countries.

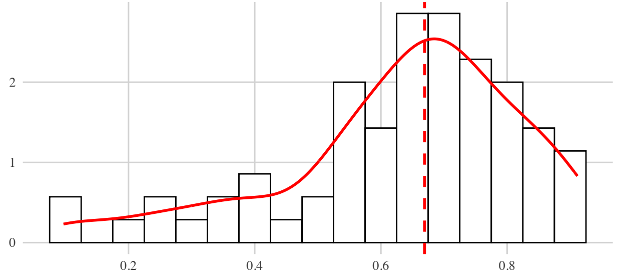


Figure. Density Inflation Weight Prediction.

Results

Regression Model

- Problem: We require weights in continuous space
- Adjustments:
  1. NKM: redefine the inflation weight: $\Omega_\pi \in [0.1, 0.9]$
  2. Simulate the NKM in 0.1 Ω_π increments
  3. Regression NN
- Repeat Training and evaluation of NN
- (As expected:) biased weight (0.67) favors the high-volatility countries.

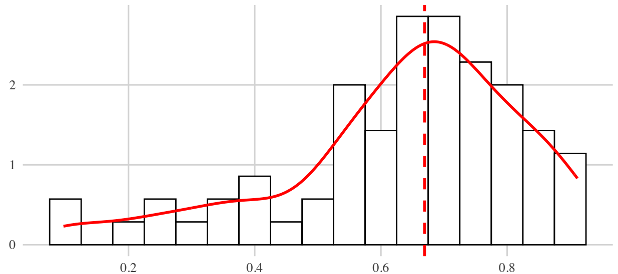


Figure. Density Inflation Weight Prediction.

Results

Regression Model

- Test our hypothesis (greater weight on greater deviation) empirically.
- OLS regression:
$$\Omega_t^H = \beta_0 + \beta_1(|\pi_t^{EMU}| - |\pi_t^L|) + \epsilon_t$$
- β_0 can be interpreted as the true weight on HV countries ω^H
- β_1 can be interpreted as ν (reaction parameter on deviations from EMU inflation)
- Expectation: $\beta_0 \approx 0.5$ and $\beta_1 > 0$ $\left(|\pi^L| \uparrow \rightarrow \ (.) \downarrow \rightarrow \ \Omega \downarrow\right)$

Results

Regression Model

- Test our hypothesis (greater weight on greater deviation) empirically.
- OLS regression:

$$\Omega_t^H = \beta_0 + \beta_1(|\pi_t^{EMU}| - |\pi_t^L|) + \epsilon_t$$

- β_0 can be interpreted as the true weight on HV countries ω^H
- β_1 can be interpreted as ν (reaction parameter on deviations from EMU inflation)
- Expectation: $\beta_0 \approx 0.5$ and $\beta_1 > 0$ $\left(|\pi^L| \uparrow \rightarrow \ (.) \downarrow \rightarrow \ \Omega \downarrow\right)$

Results

Regression Model

- Test our hypothesis (greater weight on greater deviation) empirically.
- OLS regression:
$$\Omega_t^H = \beta_0 + \beta_1(|\pi_t^{EMU}| - |\pi_t^L|) + \epsilon_t$$
- β_0 can be interpreted as the true weight on HV countries ω^H
- β_1 can be interpreted as ν (reaction parameter on deviations from EMU inflation)
- Expectation: $\beta_0 \approx 0.5$ and $\beta_1 > 0$ $\left(|\pi^L| \uparrow \rightarrow \ (.) \downarrow \rightarrow \ \Omega \downarrow\right)$

Results

Regression Model

- Test our hypothesis (greater weight on greater deviation) empirically.
- OLS regression:

$$\Omega_t^H = \beta_0 + \beta_1(|\pi_t^{EMU}| - |\pi_t^L|) + \epsilon_t$$

- β_0 can be interpreted as the true weight on HV countries ω^H
- β_1 can be interpreted as ν (reaction parameter on deviations from EMU inflation)
- Expectation: $\beta_0 \approx 0.5$ and $\beta_1 > 0$ $\left(|\pi^L| \uparrow \rightarrow \ (.) \downarrow \rightarrow \ \Omega \downarrow\right)$

Results

Regression Model

- Test our hypothesis (greater weight on greater deviation) empirically.
- OLS regression:

$$\Omega_t^H = \beta_0 + \beta_1(|\pi_t^{EMU}| - |\pi_t^L|) + \epsilon_t$$

- β_0 can be interpreted as the true weight on HV countries ω^H
- β_1 can be interpreted as ν (reaction parameter on deviations from EMU inflation)
- Expectation: $\beta_0 \approx 0.5$ and $\beta_1 > 0$ $\left(|\pi^L| \uparrow \rightarrow\ (.) \downarrow \rightarrow\ \Omega \downarrow\right)$

Results

Regression Model

Table. Main Regression Results.

	Dependent variable:				
	Inflation weight := Ω_t				
	(1)	(2)	(3)	(4)	(5)
HICP ($= \nu$)	25.09***				24.06**
	(9.41)				(9.56)
Y		3.23**			3.59**
		(1.36)			(1.44)
C			−1.83		−2.90
			(2.60)		(2.73)
L				8.95	7.44
				(6.51)	(6.29)
Constant (= ω^k)	0.62***	0.62***	0.64***	0.63***	0.62***
	(0.02)	(0.02)	(0.02)	(0.02)	(0.02)
Observations	70	70	70	70	70
R^2	0.09	0.08	0.01	0.03	0.21
Adjusted R^2	0.08	0.06	−0.01	0.01	0.16

Table of Contents

Conclusion

- We investigate whose inflation rates matter most for ECB's monetary policy.
- Theoretical model with different monetary policy rules as data-generating process
- Train machine learning model to separate rules
- Use machine learning model to classify historical EMU data between 2004 and 2021.
- Findings:
    1. Disproportional emphasis on high volatility countries
    2. Stronger reaction to countries whose inflation rates exhibit larger deviations from their long-term trend

Robustness Tests

1. Model Extension: Investment and Capital
2. Adjustment of Taylor Parameter
3. Inclusion of ECB Board Composition
4. Use of Inflation Expectations

→ No change in findings.

Current work:

- More comprehensive framework: Smets-Wouters-Model (2007, AER).
- NKM estimation
- ...

Literature

- ML in monetary policy (Tiffin, 2019; Hinterlang, 2020; Hinterlang and Hollmayr, 2021; Paranhos, 2021; Doerr et al., 2021; Fouliard et al., 2021)
- Assessment of inflation differentials within New Keynesian models (Canzoneri et al., 2006; Angeloni and Ehrmann, 2007; Andres et al., 2008; Duarte and Wolman, 2008; Rabanal, 2009; Neyer and Stempel, 2022)

Pagan frontier

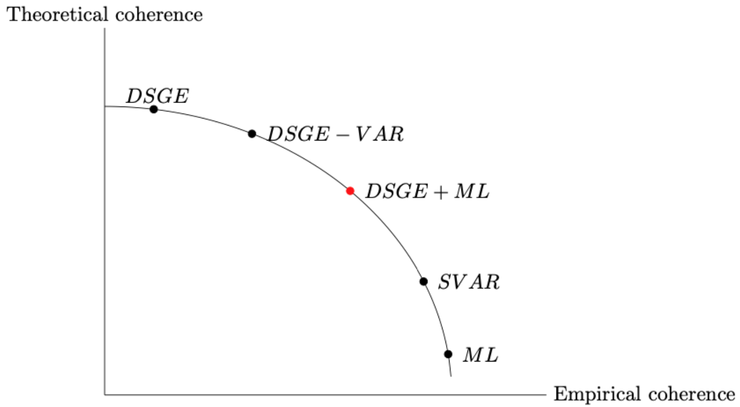


Figure. The Pagan frontier

→ we propose a modification to the Pagan frontier by combining DSGE and machine learning models to study inflation dynamics in the EMU.

→ back

Calibration

	Description	Value	
	Households		
		H	L
Ψ_k	Habit parameter	0.77	0.71
φ_k	Inverse Frisch elasticity	2.01	2.73
η_Z^k	Preference shock strength	1	0.45
γ_k	Weight of domestic goods	0.75	0.75
ϑ_C^k	Elasticity of substitution between domestic and foreign goods	1.42	1.50
ϵ	Price elasticity of demand	6	6
β	Discount rate	0.995	0.995
	Firms		
		H	L
α_k	Output elasticity labor	0.33	0.33
η_A^k	Cost-push shock strength	1	0.45
λ_k	Calvo parameter	0.737	0.852
	Central Bank		
ϕ_π	Taylor rule coefficient	1.5; 2.5	
ω_π	HICP inflation weight	$\frac{C_{SS}^H}{C_{SS}^H+C_{SS}^L}$; $[0.1, 0.9]$	

→ back

Historical EMU Data

- Data: Quarterly consumption, employment, output and price level → consumption weighted
- EMU wide interest rate → MRO + Wu and Xia (2020) shadow rate
- NKM reports percentage deviations from steady state → Hamilton (2018) filter to extract the cyclical component

→ Classification of historical inflation weight on a quarterly basis between 2004Q4 and 2022Q1.

→ Time Series → back

- We compare the performance of several algorithms in a horserace-style assessment
- All models have the following structure where $y \in (\omega_H, \omega_L, \omega_C)$ and $X \in (Y, C, \pi, ...)$:

$$y_t = h_\beta(X_t) + \epsilon_t$$

- Accuracy of models is assessed out-of-sample.
- The NN outperforms the other models by quite a margin.

→ Confusion Matrix → NNs in a nutshell

- Next: Use NN on historical EMU data

→ EMU Data

→ back

Table. Out-of-sample evaluation.

	Accuracy
Uninformed guess	0.33
MLR	0.34
Ridge regression	0.33
Lasso regression	0.33
Elastic net	0.33
K-nearest-neighbor	0.38
Decision tree	0.48
Complex tree	0.48
Prune tree	0.48
Prune complex tree	0.48
Random forest	0.67
Neural network	0.97

- We compare the performance of several algorithms in a horserace-style assessment
- All models have the following structure where $y \in (\omega_H, \omega_L, \omega_C)$ and $X \in (Y, C, \pi, \ldots)$:

$$y_t = h_\beta(X_t) + \epsilon_t$$

- Accuracy of models is assessed out-of-sample.
- The NN outperforms the other models by quite a margin.

→ Confusion Matrix → NNs in a nutshell

- Next: Use NN on historical EMU data

→ EMU Data

Table. Out-of-sample evaluation.

	Accuracy
Uninformed guess	0.33
MLR	0.34
Ridge regression	0.33
Lasso regression	0.33
Elastic net	0.33
K-nearest-neighbor	0.38
Decision tree	0.48
Complex tree	0.48
Prune tree	0.48
Prune complex tree	0.48
Random forest	0.67
Neural network	0.97

→ back

- We compare the performance of several algorithms in a horserace-style assessment
- All models have the following structure where $y \in (\omega_H, \omega_L, \omega_C)$ and $X \in (Y, C, \pi, \ldots)$:

$$y_t = h_\beta(X_t) + \epsilon_t$$

- Accuracy of models is assessed out-of-sample.
- The NN outperforms the other models by quite a margin.

→ Confusion Matrix → NNs in a nutshell

- Next: Use NN on historical EMU data

→ EMU Data

→ back

Table. Out-of-sample evaluation.

	Accuracy
Uninformed guess	0.33
MLR	0.34
Ridge regression	0.33
Lasso regression	0.33
Elastic net	0.33
K-nearest-neighbor	0.38
Decision tree	0.48
Complex tree	0.48
Prune tree	0.48
Prune complex tree	0.48
Random forest	0.67
Neural network	0.97

- We compare the performance of several algorithms in a horserace-style assessment
- All models have the following structure where $y \in (\omega_H, \omega_L, \omega_C)$ and $X \in (Y, C, \pi, \ldots)$:

$$y_t = h_\beta(X_t) + \epsilon_t$$

- Accuracy of models is assessed out-of-sample.
- The NN outperforms the other models by quite a margin.

→ Confusion Matrix → NNs in a nutshell

- Next: Use NN on historical EMU data

→ EMU Data

→ back

Table. Out-of-sample evaluation.

	Accuracy
Uninformed guess	0.33
MLR	0.34
Ridge regression	0.33
Lasso regression	0.33
Elastic net	0.33
K-nearest-neighbor	0.38
Decision tree	0.48
Complex tree	0.48
Prune tree	0.48
Prune complex tree	0.48
Random forest	0.67
Neural network	0.97

- We compare the performance of several algorithms in a horserace-style assessment
- All models have the following structure where $y \in (\omega_H, \omega_L, \omega_C)$ and $X \in (Y, C, \pi, ...)$:

$$y_t = h_\beta(X_t) + \epsilon_t$$

- Accuracy of models is assessed out-of-sample.
- The NN outperforms the other models by quite a margin.

→ Confusion Matrix → NNs in a nutshell

- Next: Use NN on historical EMU data

→ EMU Data

→ back

Table. Out-of-sample evaluation.

	Accuracy
Uninformed guess	0.33
MLR	0.34
Ridge regression	0.33
Lasso regression	0.33
Elastic net	0.33
K-nearest-neighbor	0.38
Decision tree	0.48
Complex tree	0.48
Prune tree	0.48
Prune complex tree	0.48
Random forest	0.67
Neural network	0.97

EMU time series

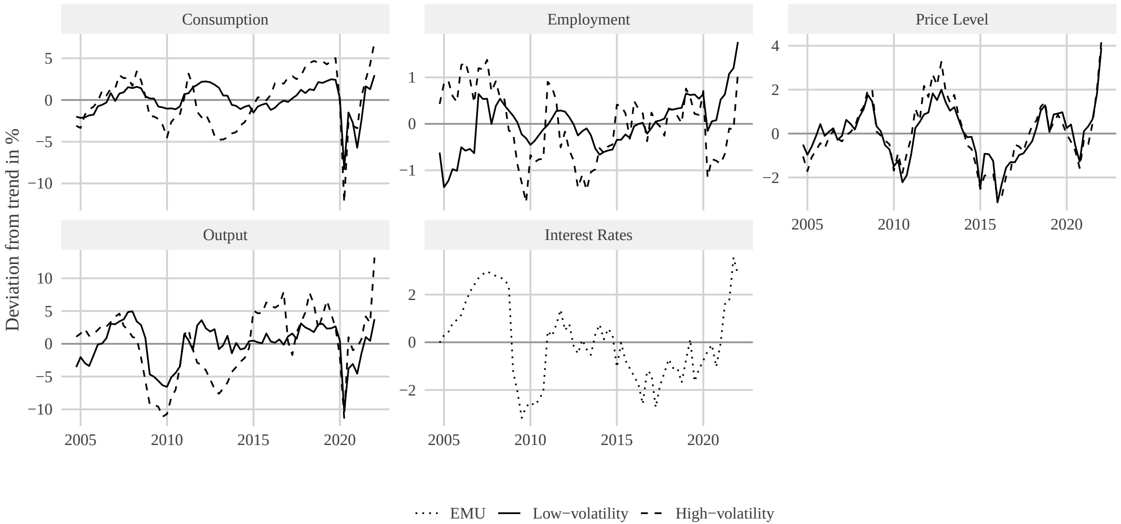


Figure. Hamilton-Filtered Data.

→ back

Model fit

Table. Comparison of Simulated Moments with Data.

Variable	Description	$\omega_\pi = \frac{C_{SS}^H}{C_{SS}^H + C_{SS}^L}$	$\omega_\pi = 0.8$	$\omega_\pi = 0.2$	Data
C_{SS}^H / C_{SS}^L	Relative consumption per capita H, L	0.962	0.962	0.962	0.805
$Y_{H,SS}/Y_{L,SS}$	Relative GDP per capita H, L	0.980	0.980	0.980	0.773
$\sigma(\hat{y}_{L,t})/\sigma(\hat{y}_{H,t})$	Relative volatility GDP L, H	0.779	0.773	0.783	0.587
$\sigma(\hat{y}_t)/\sigma(\hat{y}_{H,t})$	Relative volatility union-wide GDP, H	0.857	0.888	0.862	0.671
$\sigma(\hat{y}_t)/\sigma(\hat{y}_{L,t})$	Relative volatility union-wide GDP, L	1.010	1.149	1.010	1.144
$\sigma\left(\hat{c}_t^L\right)/\sigma\left(\hat{c}_t^H\right)$	Relative volatility consumption L, H	0.152	0.149	0.158	0.559
$\sigma\left(\hat{n}_t^L\right)/\sigma\left(\hat{n}_t^H\right)$	Relative volatility labor L, H	0.779	0.773	0.783	0.718
$\sigma\left(\hat{\pi}_t^{C,L}\right)/\sigma\left(\hat{\pi}_t^{C,H}\right)$	Relative volatility inflation L, H	0.913	0.921	0.904	0.842
$\rho(\hat{y}_{L,t}, \hat{y}_{H,t})$	Correlation GDP L, H	0.859	0.844	0.871	0.591
$\rho\left(\hat{\pi}_t^{C,L}, \hat{\pi}_t^{C,H}\right)$	Correlation inflation L, H	0.931	0.990	0.991	0.989
$\rho\left(\hat{c}_t^L, \hat{c}_t^H\right)$	Correlation consumption L, H	0.603	0.536	0.640	0.636
$\rho\left(\hat{n}_t^L, \hat{n}_t^H\right)$	Correlation labor L, H	0.859	0.844	0.871	0.132
$\rho\left(\hat{n}_t^H, \hat{c}_t^H\right)$	Correlation labor, consumption H	0.943	0.942	0.944	0.627
$\rho\left(\hat{n}_t^L, \hat{c}_t^L\right)$	Correlation labor, consumption L	0.482	0.437	0.513	0.466

Note: $\hat{x}_t$ denotes the deviation of a variable X from its zero inflation steady state.

→ back

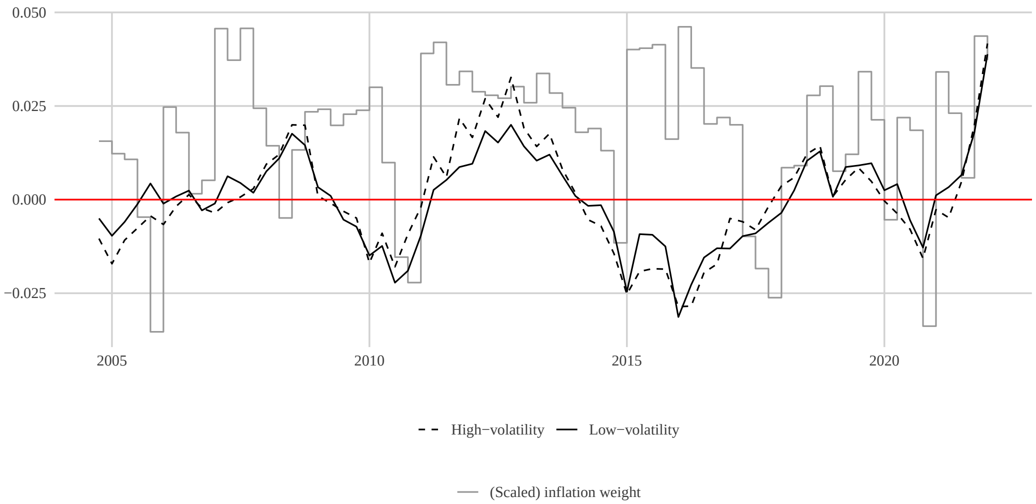
Neural networks in a nutshell II

- A neural network consists of $i \in I$ layers, with each k perceptrons.
- The input for to layer:
$$X_i = f(W_i \times X_{i-1} + b_i)$$
- Two activation functions $f(\cdot)$ in this paper::
$$f(x) = max(0, x) \qquad \text{ReLu}$$
$$f(x) = \frac{e^{x_k}}{\sum_{k=1}^{K} e^{x_k}} \qquad \text{Softmax}$$
- Training process: optimize W_i and b_i

→ back

Regression Model III

Figure. Inflation Weight from Regression NN 2004Q4 - 2022Q1.



→ Stronger deviations *L* coincide with periods of higher (*L*) weight, e.g. 2011, 2017/18 and vice versa.

→ back

Neural networks in a nutshell

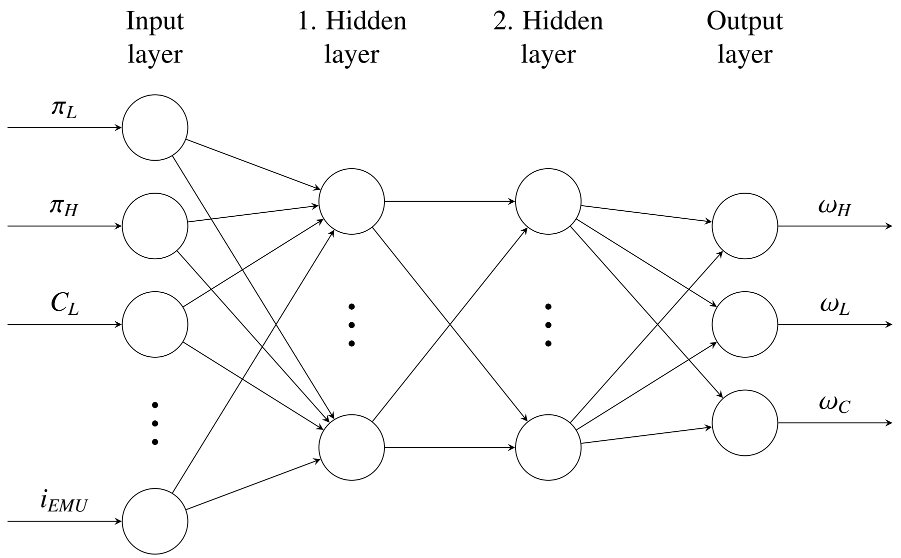


Figure. Illustration of a Neural Network.

Neural networks in a nutshell

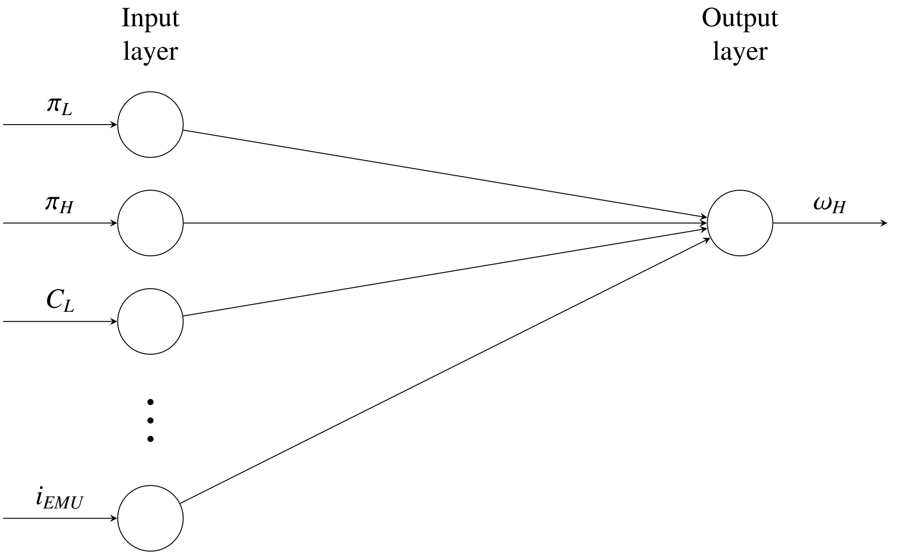


Figure. Illustration of a Logistic Regression.

Neural networks in a nutshell

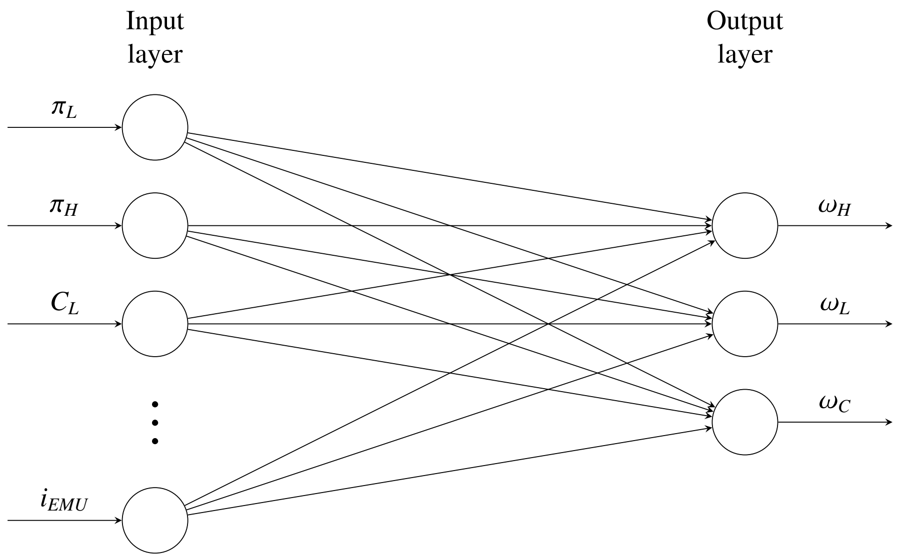


Figure. Illustration of a Multinomial Logistic Regression.

Neural networks in a nutshell

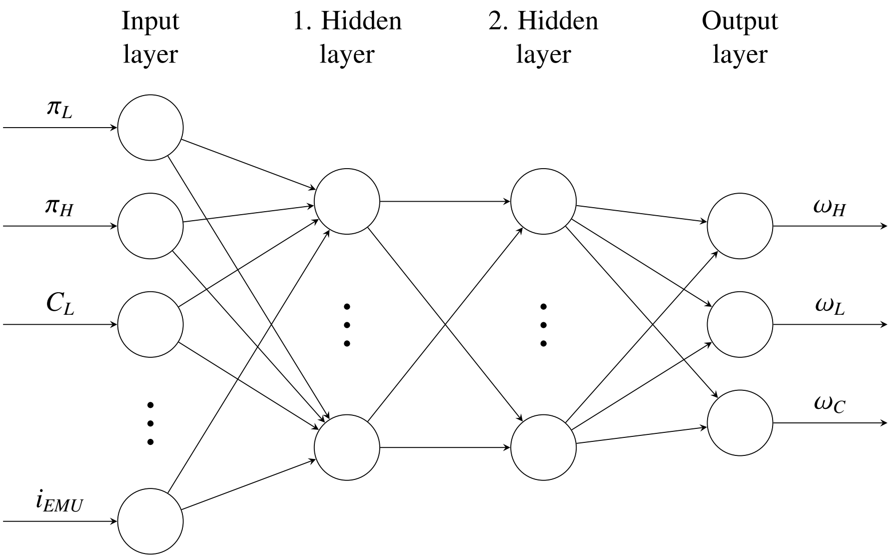


Figure. Illustration of a Neural Network.

Notes: This figure illustrates the model architecture of a feed-forward NN with four layers: One input layer, two hidden layers, and an output layer. The connections between the layers represent the weighting matrix W_i and are adjusted during the training process.

→ back → NN math

Evaluation II

Table. Confusion matrix of out-of-sample prediction by NN

		True label		
		Neutral	LV	HV
Prediction	Neutral	2405	50	39
	LV	48	2442	9
	HV	47	7	2452

→ neural network does not suffer from biased predictions

→ back